Chapter No. 6
"Flash Decorations: Flashy Templates, Headers, Banners, and Tickers"
In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.6 "Flash Decorations: Flashy Templates, Headers, Banners, and Tickers"

A synopsis of the book’s content

Information on where to buy this book

About the Author

**Suhreed Sarkar**—an IT consultant, a trainer, and a technical writer—after completing Marine engineering, he served on board a ship for two years and then started his journey into the world of IT with an MCSE in Windows NT 4.0 track. Later, he studied business administration and earned an MBA from the University of Dhaka. He has a bunch of BrainBench certifications on various topics including PHP4, Project Management, RDBMS Concepts, E-Commerce, Web Server Administration, Internet Security, Training Development, Training Delivery and Evaluation, and Technical Writing.

As a trainer, he has taught courses on system administration, web development, e-commerce, and MIS. He has consulted for several national and international organizations including the United Nations and has helped clients in building and adopting their web portals, large scale databases, and management information systems. At present, he is working on building a framework for the education sector MIS and is promoting the use of ICTs in education.

Suhreed is a renowned technical author in Bengali having dozens of books published on subjects covering web development, LAMP, networking, and system administration. He authored the *Zen Cart: E-commerce Application Development*, and *Joomla! E-commerce with VirtueMart* books. Both of them are published by Packt Publishing.

While not busy with hacking some apps, blogging on his blog ([www.suhreedsarkar.com](http://www.suhreedsarkar.com)), reading the philosophy of Bertrand Russell or management thoughts of Peter F. Drucker, he likes to spend some special moments with his family. Suhreed lives in Dhaka, Bangladesh.

Joomla! with Flash

Joomla! is an award winning Content Management System with lots of features. It is useful for almost all types of websites. It has an object-oriented, modular architecture with lots of extensibility features. You can build almost any kind of a website using Joomla!. It has a strong security, authentication, and authorization framework, a content management framework, and a templating system.

There are thousands of Joomla! components, modules, and plugins to extend the functionality of Joomla!. A component for Joomla! can be compared to an application for Joomla!. Similarly, modules are extensions to display the data on Joomla!-based websites. Plugins are also extensions for performing special tasks, such as pre- and post-filtering of content for Joomla!-based websites. With all these features of Joomla! that provide ease of administration and options for customization, sometimes you may think of integrating other technologies, such as Flash, into it. This is because Flash provides great features for animation, which is otherwise not possible using only Joomla!, HTML, or JavaScript.

Flash is unique in creating animated objects, and this adds an extra flavor to your Joomla!-based website. Besides stunning graphics, design, and layout, Flash can definitely add interactivity and value to the website. Some of the features, such as animations, small videos, and interactive games, are only possible through Flash. Considering these unique features of Flash, Joomla! developers have long been trying to use Flash content in Joomla! websites. This book shows you how to use Flash objects with Joomla! content with minimal efforts and maximum output.

What This Book Covers

Chapter 1: Get Started with Flash in Joomla! begins with a discussion on why one should opt for using Flash in Joomla! and what are the benefits of doing this. Next, it introduces us to the different types of Flash objects and various tools that will be useful for developing Flash and Joomla!-based sites.

Chapter 2: Enhance Your Joomla! Content with Flash teaches us how to use Flash-based image slideshows in our Joomla! website and display them at different module positions. As the chapter progresses, we learn to embed Flash animations into articles and finally learn to embed Flash movies at different module positions and within articles.

Chapter 3: Creating Attractive Menus with Flash begins with an introduction to the built-in menu system of Joomla!. It then illustrates the use of two extensions, Flash Floating Menu and Super Web Flash module for Joomla! 1.5, for embedding Flash-based menus into our Joomla! website and also lists a few other extensions.

In Chapter 4: Creating Flash Photo Galleries, we are introduced to the concept of building Flash photo galleries for our Joomla!-based website. This chapter begins with an illustration of the Expose Flash Gallery component and proceeds to the use of the Expose Scroller module and the Expose plugin. Next, we learn to embed a 3D Flash-based photo gallely in our website with the help of Ozio Gallery. We also learn how to embed images from Flickr, as well as from our web server, and which of the skins available with Ozio Gallery can be used for doing so. Finally, we learn to use three simpler extensions for embedding Flash photo galleries—New Gallery, Simple Image Flash Gallery, and Dynamic Flash Gallery.

Chapter 5: Flashier than Ever: Maps, Charts, Custom Fonts, Multimedia, and More extends the use of Flash on our Joomla!-based website. Firstly, we learn to build interactive Flash-based maps and charts using the YOS amMap and amChart components respectively. Then the chapter illustrates the use of sIFR with Joomla! for displaying the contents of a Joomla! site in fonts that are not installed on users’ computers. The use of the Joomla! Flash Uploader component is also illustrated. Further, the chapter takes us through how to create streaming video sites with the use of the JVideol components and discusses various extensions for adding an MP3 player on your Joomla! website.

Chapter 6: Flash Decorations: Flashy Templates, Headers, Banners, and Tickers deals with using Flash templates for decorating our website with Flash logos, headers, banners, and tickers. In this chapter, we are introduced to the two ways of showing Flash objects in a Joomla! site, namely, by embedding the Flash object in a Joomla! template showing it permanently on a Joomla! website and by using a suitable module. Furthermore, the use of Joomla! modules like FlexHeader3 and Web Flash Joomla! for displaying Flash objects is also discussed in this chapter.

Chapter 7: Playing with Code focuses mainly on the issue of how we embed Joomla! content into a Flash site, which is exactly the opposite of what has been done in the previous chapters, using the J-AMFPHP component. In this chapter, we also take a look at some of the accessibility issues and ways to address these issues.

Chapter 8: Troubleshoot Your Applications deals with the most common problems with Joomla!, Flash, and the Joomla! extensions that have been discussed in this book. Besides pointing to the probable solution for such problems, the chapter also provides references to relevant websites and forums that can prove to be helpful in solving problems.

Appendix: Resources for Joomla! and Flash gives you a list of some more resources that can be useful for using Flash with Joomla!. Firstly, it shows some resources for Joomla!, and then it lists some Flash extension-specific resources.

Flash Decorations: Flashy Templates, Headers, Banners, and Tickers

So far we have seen how to use Flash for enhancing our content by adding Flash animations, charts, maps, movies, videos, and sound clips. We have not yet used it to change the decoration of our site though. In this chapter, we are going to use Flash for decorating our site. On completion of this chapter you will be able to use:

- Flash-based templates for your Joomla! website
- Flash logos
- Flash headers
- Flash banners

In this chapter, we will mainly focus on the visual design of our site. To acquire the information presented here, it is assumed that you have some basic understanding of Joomla!’s visual design including templates, components, module position, and so on.

Adding Flash in templates

If you are familiar with Joomla! templates, then you will understand that there are two ways to display Flash in a template:

- By hardcoded embedding of Flash items
- By dynamically loading Flash objects at module positions

We have seen many modules that can display Flash objects. Therefore, in this section, we will be looking into the embedding of Flash objects within templates. It will also be helpful if we understand the structure of Joomla! templates.

Generally templates for Joomla! include headers in Flash. Flash animations are included in the header area of a Joomla! template. Some templates include the mechanism to show images from a specific directory. For example, the template shown in the following screenshot, available for download at http://joomlatp.com/joomla-1.5-templates/Templates-has-flash-header.html, is designed to show a Flash header comprised of the images kept in a directory:

![Flash header](image)

The following sections briefly describe the structure of a Joomla! template and the ways to embed a Flash object in this template.

## Structure of a Joomla! template

The look and feel of Joomla! is determined by templates. You can apply a template to the frontend as well as to the backend. Templates for the Joomla! frontend reside in the `/templates` directory of the Joomla! webroot, while those for the administration panel are found in the `/administrator/templates` directory. You can install multiple templates and apply one or more templates to the different sections. However, you must designate one default template for the site.
To designate a default template, go to Extensions | Template Manager. Select the desired template and click on the Default button on the toolbar. For assigning a template to a specific section of the site, click on a template, except the default template, and then select the section or the menu item for which you want to assign the template from the Menu Assignment section.

If you examine the directory structure of a Joomla! template, you will find at least the following subdirectories in the templates directory:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mx_joofree2</td>
<td>This is the main template directory. It contains some subdirectories and at least the following files under its root:</td>
</tr>
<tr>
<td></td>
<td>• index.php: This is the main file for a template. The basic structure of a Joomla! template is defined in this file. We will examine this file later.</td>
</tr>
<tr>
<td></td>
<td>• templateDetails.xml: This XML file defines the template by mentioning its designer, the different files bundled with it, the positions and parameters available, and so on.</td>
</tr>
<tr>
<td></td>
<td>• params.ini: This file contains the parameters and their default values. For example, a template may use several colors for theming, but users can select a preferred color as a parameter for this template, and that information is stored in this file.</td>
</tr>
<tr>
<td>mx_joofree2/css</td>
<td>This directory contains all the cascading stylesheets to be used for a Joomla! site. This directory will contain at least one stylesheet named template_css.css. It may also contain a stylesheet named template_ie6.css and other stylesheets.</td>
</tr>
<tr>
<td>mx_joofree2/html</td>
<td>This folder may contain some definitions for the custom rendering of certain parts of the site. For example, the mx_joofree2 template contains two files—module.php and pagination.php. These two files define custom module rendering and pagination for Joomla!. For more information on using HTML overrides, refer to <a href="http://docs.joomla.org/How_to_override_the_content_from_the_Joomla!_core">http://docs.joomla.org/How_to_override_the_content_from_the_Joomla!_core</a>.</td>
</tr>
<tr>
<td>mx_joofree2/images</td>
<td>This folder contains the images for the template. It may contain a logo image, a background image, and so on. It may also contain some subdirectories, for example, the mx_joofree2 template contains a subdirectory images/headers, where the header images for the template are stored.</td>
</tr>
</tbody>
</table>

As you know, the main structure of a Joomla! template is defined in the `index.php` file. The file looks as follows:

```php
<?php
// no direct access
defined( '_JEXEC' ) or die( 'Restricted index access' );
```

This line of code is to prevent direct access to the file. This is a convention to prevent direct access to any file in Joomla!. After this, the following lines define some variables for the template that will be used later on in the template:

```php
define( 'YOURBASEPATH', dirname(__FILE__) );
$template_path = $this->baseurl . '/templates/' . $this->template;
$live_site = $mainframe->getCfg('live_site');
$template_path = $this->baseurl . '/templates/' . $this->template;
$show_flashheader = ($this->params->get("showFlashheader", 1)  == 0)?"false":"true";
$show_logo = ($this->params->get("showLogo", 1)  == 0)?"false":"true";
$show_date = ($this->params->get("showDate", 1)  == 0)?"false":"true";
$show_breadcrumbs = ($this->params->get("showBreadcrumbs", 1)  == 0)?"false":"true";
?>
```

Having defined the template variables, the template's structure starts with some common HTML declarations as follows:

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language; ?>" lang="<?php echo $this->language; ?>">
<head>
Whatever is included after the following lines will be considered as included in the `<head>` section. Generally, stylesheets, JavaScript, and site-wide files are included in a template's `<head>` section with the following lines:

```html
<jdoc:include type="head" />
<link rel="shortcut icon" href="<?php echo $this->baseurl; ?>/images/favicon.ico" />
<link href="<?php echo $this->baseurl; ?>/templates/system/css/system.css" rel="stylesheet" type="text/css" />
<link href="<?php echo $this->baseurl; ?>/templates/system/css/general.css" rel="stylesheet" type="text/css" />
<link href="<?php echo $this->baseurl; ?>/templates/<?php echo $this->template?>/css/template_css.css" rel="stylesheet" type="text/css" />
```
```
Next, the body section of the template starts:

```html
<body class="body_bg">
<div id="bgr">

<div id="wrapper">
<div id="tophead">

The following block checks whether any module for position user4 is enabled. If the count of enabled modules is greater than zero, then a `<div>` element with an ID search is displayed, and modules specified for position user4 are displayed as children of this `<div>` element:

```html
<?php if($this->countModules('user4')) : ?>
<div id="search">
<jdoc:include type="modules" name="user4" style="xhtml" />
</div>
<?php endif; ?>
```

Having placed the modules for the position user4, the following block now defines where to show the site's logo. This block of code dynamically selects the site's logo and links that to the site's home page:

```html
<!-- BEGIN: LOGO -->
<?php if($show_logo == "true") : ?>
<div id="logo">
<a href="<?php echo $mosConfig_live_site;?>">
<img src="<?php echo $this->baseurl; ?>/templates/<?php echo $this->template?>/images/logo.png" alt="" />
</a>
</div>
<!-- END: LOGO -->
```

If we want to show a Flash header beside the logo, then we can add the following block to include a Flash object:

```html
<!-- BEGIN: flashheader -->
<?php if($show_flashheader == "true") : ?>
<div id="ol-flashheader">
```

Flash Decorations: Flashy Templates, Headers, Banners, and Tickers

This code block first checks whether the variable for showing a Flash header is checked or not. If it is checked, then a `<div>` element will be created, under which the Flash object is embedded using the `<object> </object>` tags.

The following code block is used to display the current date on the site:

```php
<?php if($show_date == "true") : ?>
    <div id="date-format">
        <?php $now = JFactory::getDate(); echo $now->toFormat("%A, %d %b %Y"); ?>
    </div>
<?php endif; ?>
```

The following block defines another module position. Modules designated for the `user3` position will be displayed here. Note that modules are included with the `<jdoc:include type="modules" name="position-name" />` markup:

```php
<?php if( $this->countModules('user3') ) {?}
    <div id="topcol">
        <div id="topmenu">
            <table cellspacing="0" cellpadding="0" style="float: left;">
                <tr>
                    <td>
                        <jdoc:include type="modules" name="user3" />
                    </td>
                </tr>
            </table>
        </div>
    </div>
<?php } ?>
```

The following code block displays another module called breadcrumbs. First, it checks whether the parameter is set to show breadcrumbs or not. If it is, then the breadcrumbs module will be displayed in the block:

```php
<?php if ($show_breadcrumbs == "true") : ?>
    <div class="path">
        You are here: <jdoc:include type="module" name="breadcrumbs"/>
    </div>
<?php endif; ?>
```

Next comes the module for the left position:

```php
<?php if($this->countModules('left') and JRequest::getCmd('layout') != 'form') : ?>
    <div id="left">
        <jdoc:include type="modules" name="left" style="rounded"/>
    </div>
<?php endif; ?>
```

The following block includes the modules designated for the right position:

```php
<?php if($this->countModules('right') and JRequest::getCmd('layout') != 'form') : ?>
    <div id="main">
<?php else: ?>
    <div id="main_full">
<?php endif; ?>
    <div class="nopad">
        <jdoc:include type="message"/>
    </div>
<?php endif; ?>
```

The next code block counts the number of modules available for the user1 and user2 positions and then includes the modules designated for these two positions. Note the highlighted line of code in this code block. It shows the component currently selected:

```php
<?php if($this->countModules('user1') ) {?}
    <td class="lr-padd">
        <jdoc:include type="modules" name="user2" style="xhtml"/>
    </td>
<?php endif; ?>
```

Finally, here comes the footer section. The highlighted line in the following code block includes a footer file to show the footer text:

```php
<?php include (dirname(__FILE__).DS.'/footer.php');?>
```

All this code in the index.php file when rendered with template_css.css and other stylesheets will display a layout like the one shown in the following screenshot:

In a Joomla! template, the site's data comes from the database and fits in different positions. When you are planning for Flash-based Joomla! templates, you should first identify where to put the Flash objects and where to display the dynamic data for the site. In the template layout, we have seen that a suitable place for adding a Flash object are the logo and the header positions. We can also use some extensions to display Flash headers. The following sections explain how to use extensions for showing Flash logos and headers.

**Using Flash logos**

A logo is a special graphic displayed at a specific position on the Joomla! template. You can replace the ordinary image with Flash-animated logos by embedding it in that position. As we have seen earlier, Flash objects are embedded in Joomla! templates using the `<object>` tag. For styling purposes, you may put this element under a `<div>` element, as shown in the following code:

```html
<div id="logo">
    <object type="application/x-shockwave-flash"
            data="/templates/mx_joofree2/images/logo.swf"
            width="200" height="200">
        <param name="movie" value="/templates/mx_joofree2/images/logo.swf">
    </object>
</div>
```

As you can see, the `<object> </object>` tag can contain child elements. We pass parameters to the Flash objects using the `<param>` element. For example, we have passed the name of a movie file using the highlighted tag in the previous code.

Some of you may already know about the SWFObject JavaScript library. It is used to embed Flash files using JavaScript. You can also use it to embed Flash objects in Joomla! and can also generate Flash objects. It is freely available at http://code.google.com/p/swfobject/. Brief documentation on its usage is also available there.

### Using Flash headers

We have seen that one of the uses of Flash in Joomla! templates is as a header. By using a Flash animation in a site's header you can create some stunning effects. As we have already seen, while designing the template, we may embed Flash animation in the header region and control the layout using an appropriate CSS stylesheet. To embed such Flash animations like these, you can use the `<object> </object>` XHTML tag. We have seen its use in the previous section. An alternative to this is showing the Flash header at some module position. There are several extensions that can be used for showing Flash objects at a module position. We will be looking at some of them next.

### Using Flexheader3

Flexheader3 is a Joomla! 1.5-compatible extension for using Flash as headers in Joomla! sites. This is available for download for free at http://flexheader2.andrehotzler.de/en/download/folder/208-flexheader3.html. After downloading the package, install it from the Extensions | Install/Uninstall screen in Joomla! administration. Then click on Extensions | Module Manager. In the Module Manager screen, you will find the module named Flexheader3. Click on it and that shows the Module: [Edit] screen for the Flexheader3 module, as shown in the following screenshot:
Chapter 6

The Details section is similar to other modules from where you enable the module, select the module position to display this, select the order of display, and assign menus for which this module will be displayed. The module-specific settings are in the Parameters section. As you see, selecting the module position is crucial for this module. Most of the templates don’t have a position to display the header using a module. Therefore, you may need to create a module position for displaying a Flash header. The following section shows you how to create a module position displaying a header.

Creating a module position

To create a module position in your template you need to edit at least two files. Browse to the /templates directory, and click on the name of the template that you want to modify. You need to edit two files in the template folder: index.php and templateDetails.xml. First, open the templateDetails.xml file in your text editor and find the <positions> tag. Under this, type the line highlighted in the following code so that the file looks like the following:

```
<positions>
   <position>flexheader</position>
   <position>left</position>
   <position>user1</position>
   ...
   <position>right</position>
   <position>debug</position>
</positions>
```

Remember to type `<position>flexheader</position>` before ending `<positions>` tag. Placing it outside the `<positions> </positions>` block will make the template unusable.

After modifying the templateDetails.xml file, open the index.php file in your text editor. Find out the code for including a header image in that template. Generally, this is done by inserting an image using the `<img src=... />` tag. If you don't find such a tag, then look for `<div id="header" ...>` or something like that. In such cases, CSS is used to display the background image to the `div` element.

Once you have found the code for showing the header image, replace it with the following code:

```
<jdoc:include type="modules" name="flexheader" style="RAW" />
```

This line of code means that you are instructing to include modules designated for the `flexheader` position. When we assign the Flexheader3 module to this position, the contents of that module will be displayed in this position. Generally, this module will produce a code like the following in this position:

```
<img src="/images/header.png"
     title="My header image"
     alt="Header image"
     style="width: 528px; height: 70px;"
 />
```

When changes to `index.php` are made, save those changes. We will be configuring the module to display a Flash header in this module position.

**Configuring the module**

After creating a module position for Flexheader by modifying the `index.php` and `templateDetails.xml` files, you can configure the Flexheader3 module to show the Flash header. To configure the Flexheader3 module, go to the Parameters section in the Module: [Edit] screen for the Flexheader3 module. Now we will look into the Module Parameters section shown in the following screenshot:

In the **Module Parameters** section you need to configure the following:

- **Show Help information on Frontpage**: When showing Flash headers you may display some help tips for the frontend users. You can choose:
  - `none` — to prevent showing such help
  - `for endusers` — if you want to show help to site visitors
  - `for developers` — if you want to show help messages to developers only.

  Selecting `for developers` is logical as that enables the messages and fixing if needed.

- **Location of image folder**: Select the path of the folder where you have stored the images to be displayed in the header. You can select `/images` or `/templates/current template`. Selecting `/images` stores the header images in the global `/images` directory, which can then be accessed through the media manager. On the other hand, you can store the images into the template's `images` folder. In such a case, select the `/templates/current template/` option from the drop-down list.

---

• **Name of image folder:** Specify the name of the folder where the header images will be stored. If you have selected /images in the previous field and type **header-images** in this field, then the images will be stored in the /images/header_images folder.

• **Name of default image (without extension):** Type the name of the default image to be displayed if there is no image. Specify the name without the file extension; extensions for the images will be defined later.

• **Filetype of image:** Specify the type of images used for the header. You need to select an extension from the drop-down list. The available extensions are: **jpg, jpeg, gif,** and **png.** Remember that when you select an extension the images to be used for the header have to be of that type. If you select **png,** for example, the header images need to be in the PNG format.

• **Height:** Specify the height of the header images in pixels or percentage, as indicated in the **Indication of Dimensions** field. All images need to be resized to this height.

• **Width:** Specify the width of the header images in pixels or percentage, as indicated in the **Indication of Dimensions** field. All images need to be the same width as this.

• **Indication of Dimensions:** Select either **px** or **%** to indicate the unit of dimension. For example, if you select **px** in this drop-down list, the values in the **Height** and **Width** fields will be in pixels. For a fluid layout you may specify the image height and width in percentage. In this case, you have to select **%** from the drop-down list.

You can display images, XHTML `<div>`, or Flash objects using the Flexheader3 module. The **Advanced Parameters** section for the module, shown in the next screenshot, allows you to configure what will be displayed in this module and how:

---

### Module Parameters

#### Advanced Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display mode</td>
<td>Image</td>
</tr>
<tr>
<td>DIV Content (for Display mode=DIV Area with background image)</td>
<td></td>
</tr>
<tr>
<td>IMG Alternative Text (XHTML ALT)</td>
<td></td>
</tr>
<tr>
<td>Hyperlinked header</td>
<td></td>
</tr>
<tr>
<td>Hyperlink Target</td>
<td>/</td>
</tr>
<tr>
<td>JoomFish Support</td>
<td></td>
</tr>
<tr>
<td>VirtueMart Support</td>
<td></td>
</tr>
<tr>
<td>Load a css file</td>
<td></td>
</tr>
<tr>
<td>Position of Debug information</td>
<td>On bottom of frontpage</td>
</tr>
<tr>
<td>Debug Background Color</td>
<td>Red</td>
</tr>
<tr>
<td>Debug Opacity</td>
<td>80</td>
</tr>
<tr>
<td>Debug Height</td>
<td>auto</td>
</tr>
<tr>
<td>Module Class Suffix</td>
<td></td>
</tr>
<tr>
<td>Flexheader3 CSS Class</td>
<td></td>
</tr>
<tr>
<td>Caching</td>
<td>Use Global</td>
</tr>
<tr>
<td>Cache Time</td>
<td>900</td>
</tr>
</tbody>
</table>

---

From the Advanced Parameters section you need to configure the following:

- **Display Mode**: Select a display mode from this drop-down list. You can either display an image, a `<div>` object, or a Flash object with this module. Select any one from the drop-down list: **Image**, **DIV area with image as background**, or **Flash Object**. One of the reasons you may want to use `<div>` is that you want to show some text in the header. When you use a `<div>`, images are used as background to the `<div>` tag and the text inside the `<div>` tag is visible. Based on the selection of this field, you need to configure the other fields.

- **DIV Content (for Display mode=DIV Area with background image)**: When you have selected **DIV Area with background image** in the Display Mode field, you need to specify the DIV content in this textbox. For example, if you want to display the logo in the header, then simply type `<img src="/templates/mx_joofree2/images/logo.png" />` and the logo will be displayed as an overlay to the header images. However, for formatting the DIV content, you also need to specify the relevant styles in the CSS file.

- **IMG Alternative Text (XHTML ALT)**: When you select **Image** in the Display Mode field to display images in the Flexheader3 module, you can specify alternative text for images, which is equivalent to the `alt` attribute in the `<img>` tag.

- **Hyperlinked header**: You can make the header hyperlinked. If you want to make it hyperlinked, then select **Yes** for this field.

- **Hyperlink Target**: If you have selected **Yes** for the **Hyperlinked header** field, then specify the URL for the hyperlink in this field. This URL may be relative or absolute.

- **JoomFish Support**: Flexheader3 can display language-specific images. For example, your site has two languages—English (en), and French (fr). You create the language-specific header images, such as `1-en.png`, `2-en.png`, `3-en.png`, `4-en.png`, and so on for English, and `1-fr.png`, `2-fr.png`, `3-fr.png`, `4-fr.png`, and so on for French. Then, to enable the multilingual feature, select **Yes** in the **JoomFish Support** field.

- **VirtueMart Support**: Prefixing the image names with the VirtueMart category and product IDs, such as `category1_product1_01.png`, may link that image to that category and product. To enable this support for linking with VirtueMart products and categories, select **Yes** in this field.

- **Load a css file**: Select **Yes** to use a CSS file for formatting different pages.

---

• **Position of Debug Information**: Specify the position where you want the debug information for this Flexheader3 module to be displayed. You can show this on top or at the bottom of the front page. This will be very useful for adjusting the module's output.

• **Debug Background Color**: Specify a background color for the debug information shown on the front page. You can select Red, Blue, Yellow, Green, or Cyan as the background color for the debug information.

• **Debug Opacity**: Specify the opacity for the debug information by selecting a number from 5 to 100 from the drop-down list.

• **Debug Height**: Specify the height of the debug window as a percentage of the size of the main window. Select a percentage from the drop-down list or select *auto* to let the height be adjusted automatically.

• **Module Class Suffix**: For individual module styling, you can specify a suffix to the module class so that you can apply appropriate CSS styles to this class suffix. For example, you can specify the module suffix `myclass`, and add a CSS declaration `module-heading-myclass` to define your own style for this module's heading.

• **Flexheader3 CSS Class**: You can specify a CSS class for the Flexheader3 module. You should use such a class if you have included the Flexheader3 module as *raw*, without any styling.

• **Caching**: You can select *Use Global* from the drop-down list to enable caching for this module. Select *No caching* to disable caching.

• **Cache Time**: If you have chosen to use caching, then specify the time in minutes, after which the module will be re-cached.

If we configure these settings for the Flexheader3 module and save them, you will find the header—either images from the specified directories or SWF animations—displayed in the newly created Flexheader module position. The module will look similar to the following screenshot:

![Image](image.png)

Using Web Flash Module to show headers

We have seen the use of mod_web_flash in Chapter 3, Creating Attractive Menus with Flash. There we learned how to use this module for creating menus. In fact, we can use the same thing as a site's header. The only difference will be in configuring the module differently with appropriate images, texts, and link URLs. Then we have to publish that module in a position like the Flexheader3 module position. For example, placing the same module that we had configured in Chapter 3 would display the header, as shown in the following screenshot:

When using the Web Flash Joomla 1.5 module (mod_web_flash), you can also add menu links to the rotating header images. It will be nice to add different background images, some text (a slogan for your site), and then linking them to the different parts of your website.

Using Flash banners

Joomla! has a banner management system from where you can add banner clients, categorize the banners, define the banners, and publish the banners through the module. Usually you can display banner images using this component and also manage the banners very effectively. You can specify the banner images and links for the same. The banners can be displayed for a specified duration or for a certain number of impressions. Unfortunately, we cannot display a Flash banner using this banner component.

For adding animated Flash banners, we need to use third-party extensions. Any module that can show a Flash object at a module location can be used to display Flash banners. For example, we have already learned how to use the Web Flash Joomla! 1.5 module. We can use this module as a banner. To do this, first create a Flash animated banner and configure the module to display this Flash object at a position called banner in your template.

You can always hardcode your Flash objects to a Joomla! template. In this case, Flash banners can also be displayed in a Joomla! site using the `<object> </object>` tags. However, you will have to edit the template file each time you change the banner.

### Flash tickers

Tickers are used for displaying small amounts of information on your site. You can either use text-based tickers or animated Flash tickers. The same principle applies when displaying animated Flash tickers. If you want to display an animated Flash ticker, design the ticker with appropriate text, images, sounds, and so on, and publish it on your site. Like others, you can display this either by embedding the ticker Flash object in the template, or by publishing it through a module. If you want to use a module for publishing tickers, use the Web Flash Joomla! 1.5 module or any other module that can display one or more animated Flash objects from a directory on your web server.

### Summary

Although Flash can be used for designing the layout of a Joomla! site, it is often used as a header or a logo for the site's template. You can use Flash objects in different parts of the Joomla! template to display the Flash objects. In this chapter, we have learned two ways of showing Flash objects in a Joomla! site: by embedding the Flash object in a Joomla! template and showing it permanently on a Joomla! website; and by using a suitable module. First, we have seen how to embed Flash objects in Joomla! templates using the `<object> </object>` tag. Later, we have explored options for displaying Flash objects using the Joomla! modules such as FlexHeader3 and Web Flash Joomla!.

In the following chapter, we are going to learn about using web services with Flash and the accessibility issues with it.
Where to buy this book

You can buy Joomla! with Flash from the Packt Publishing website:

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.