/Flash/styling with CSS and Flash

On your essential CD
All the code you need to complete this tutorial is included on your disc.

Paul Wyatt shows you how to use external CSS and text files to produce interactive web content, and how to create a preloader for dynamically loading JPEG images

Knowledge needed	Basic CSS, Flash and HTML
Requires	Flash 8, a text/HTML editor
Project time	40 minutes

A few years ago, editing text was a battle in Flash. Formatting and tweaking was always done within the application itself, so Flash designers would be called on to update text boxes within their Flash masterpieces – the source file would have to be opened, the copy tweaked and styled using Flash tools, and a new SWF file exported. This became further complicated and more of a chore if you had to change images as well, as these would have to be updated in a similar fashion.

Over the last few years, there's been light at the end of the tunnel. HTML formatted text can now be loaded into a dynamic Flash text field. In Flash Player 7 and above, by using the <img> tag in a text field, you can embed a variety of image formats such as JPEG, GIF and PNG, plus SWF files and movie clips. Text will even wrap around images in the same way it does on a regular HTML page.

The crisp, clean and elegantly styled text that Flash produces can now be dynamically loaded into Flash, formatted with HTML, then styled with an external Cascading Style Sheet (CSS). The latter can be used to group style rules, which can be applied to HTML or XML elements.

Using a style sheet gives you the flexibility of separating out the styles and layout of your page – or Flash elements – from the rest of the web page data. Key styles such as margins, fonts and font sizes can be grouped together and put in one place. As the name indicates, the styles are held within one 'sheet', and can be easily modified by simply updating this one file. What's more, if you also have HTML elements to your website, you can even use the same CSS for both these and the Flash elements.

By taking advantage of the flexibility of working with external text and CSS files, you have the freedom to create fully animated and interactive site elements without worrying about updating the content manually. The FLA files can be left safely stored away to never again be accidentally messed up by some hapless soul who was "just trying to update the text".

But before you get too excited, there are a few limitations. Although Flash 8 has greater support than previous versions of Flash, it still only supports a limited number of HTML tags. These are as follows:

What you'll build Head to the .net CD to take a look at the final result of this tutorial: a Flash content block using three SWF files, which are updated via external text files

<a> - anchor
<b> - bold tag
  - break tag
<font> - Font tag
<img> - image tag
<i> - italic tag
- list item tag
<p> - paragraph tag
- span tag
<textformat> - Text format tag
<u> - underline tag

Don't be too dismayed, though. With built-in CSS support, you can redefine HTML formatting tags. You can also create 'classes', which you can apply to targeted HTML elements using the <p> or tag's class attribute.

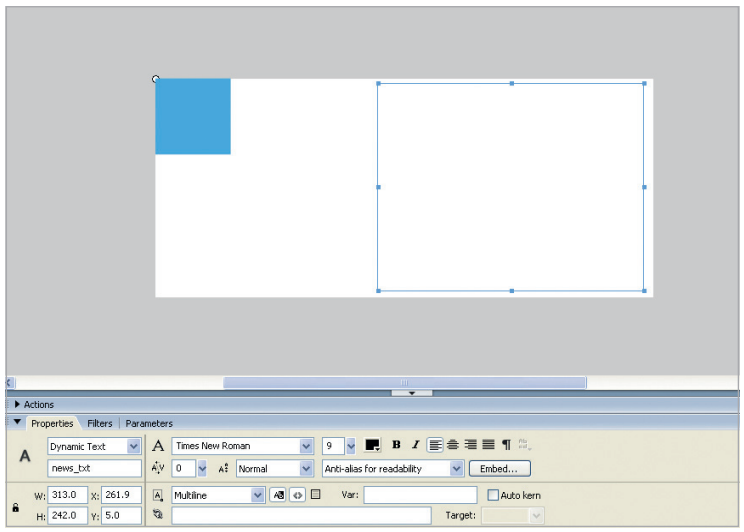
Getting started

In this tutorial, I'll walk you through this process. On the .net CD, in the 'dotnetrotator' folder, open up the 'webpage' subfolder and click the 'index.html' file to see what you'll be creating. It's a Flash content block that pulls in three different SWF files when the 'design', 'develop' and 'discover' buttons are pressed. These loaded-in SWF files have the same CSS file applied to them, and act as content containers that are updated externally via separate text files.

First, create your text file for one of the news items. This needs to be a plain text file with the '.txt' file extension. Flash can be quirky, so when you create your text, don't put any carriage returns in the file. These will create large gaps between lines when the text is eventually sent to the Flash file.

Each of the news items for the Flash content block consists of a headline, subheading, body text, links and highlighted text. At the end of each of these, we'll break them up for now, adding break <br> tags. We'll add the style elements after we've created the line breaks.

When adding HTML tags into a file that will be brought into Flash, there are a number of ground rules that you need to follow. For example, when defining



Dynamic text Use the Property Inspector to give your text field an instance name. Set the text field to 'dynamic text', and make sure that 'render text to html' and 'multiline' are set

HTML colours, always use longform hex values. Don't shorten them, or use names such as red, green or yellow. Flash is stubborn and appreciates neatness, so longhand it is! If more than one tag set defines your styled text, you need to make sure that the tags follow the "last in, first out" rule. To save you hours of trying to problem-solve, remember that attributes of HTML tags must be enclosed in single (') quotation marks. Any that aren't can cause Flash to go into a pout and refuse to render correctly. If you find that your text isn't looking as it should, be sure to check that all of these rules have been followed.

Adding a style sheet

Save the text file, then create another file (using your favoured text editor or HTML editor). Name the new file 'styles.css' – you're now ready to get styling! Our style sheet contains the following elements:

```
.headline {
  font-family: Arial ,Helvetica,sans-serif;
 font-size: 25px;
 font-weight: bold;
   }

.subHeading {
   color: #666666;
   font-size: 12px;
   }

p {
   font-family: Arial ,Helvetica,sans-serif;
   font-size: 12px;
   color: #000000;
   display: inline;
}

a:link {
   color: #000000;
   text-decoration: underline;
}

a:hover {
   color: #fc018e;
}

.emphasized {
   font-weight: bold;
   color: #fc018e;
   font-size: 15px;
}
```

>>

Making a JPEG preloader animation

Let your visitors know what's happening

Here, we're going to make an animation to illustrate how long the JPEGs in your Flash movie will take to load. Open the 'design.fla' file. The small blue box in the left-hand corner is a 100-frame movie clip called 'loader_animation'. If the JPEG has loaded 45 per cent of 30k, 45 frames will have played. When 100 per cent is loaded, it reaches frame 100 and loads the JPEG into a blank movie clip.

Create your animation, making sure you've placed a 'stop();' ActionScript on frames 1 and 100. Drag an instance of this onto the stage and give it an instance name of 'sitePreloader'. Now add the following ActionScript:

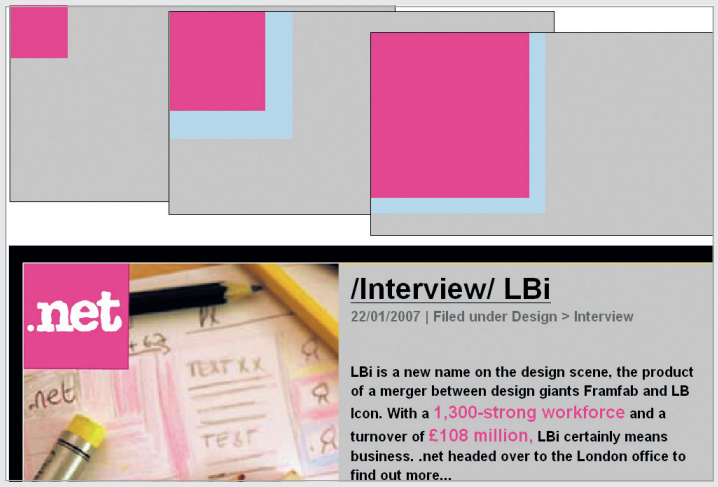
```
var myMCL:MovieClipLoader = new MovieClipLoader();
var myListener:Object = new Object();
myMCL.addListener(myListener);
myListener.onLoadProgress = function (target_mc:MovieClip, bytesLoaded:
Number, bytesTotal:Number) {
  var loaded:Number = Math.round((bytesLoaded/bytesTotal) * 100);
  sitePreloader.gotoAndStop(loaded);
}

myListener.onLoadInit = function (target_mc:MovieClip) {
  sitePreloader._visible =  false;
}

myListener.onLoadStart = function (target_mc:MovieClip) {
  sitePreloader._visible =  true;
}
myMCL.loadClip("/images/image2.jpg", "load_images");
```

This uses 'MovieClipLoader' to illustrate the status of loading data. A listener is created to process this. A calculation is produced, which is rounded up ('Math.round') and passed to the animation 'gotoAndStop(loaded)' – with 'loaded' being the percentage and the frame number for the playhead to move to in the animation. 'onLoadInit' ensures that the animation vanishes when loaded.

```
(" sitePreloader._visible =  false;")
```

Finally, the ActionScript loads the JPEG into the blank movie clip, with the instance name of 'load_images' in the 'image load' layer.

```
myMCL.loadClip("/images/image2.jpg", "load_images");
```



Preloaded Flash will play your preloader animation in stages according to how many bytes have been loaded. If 30% of the image is loaded, 30% of the animation will play

Big and bold This is the way to go with CSS! Experiment with different CSS styles until your text makes an impact and is as clear and visible as possible

>> As the content is broken up into key areas, there is a style created for each of these. It's important that the content is clear and legible, so choose styles that support this.

Each style is applied to the main text file as a <p> or class attribute. For the headline, you'll want to use the 'headline' class, but you also need to turn it into a link. So, this is defined in the text file as follows:

```
<p class='headline'><a href='http://www.netmag.co.uk/zine/design-interview/lbi'
target='_blank'>/Interview/ LBi</p></a>
```

In this instance, you're applying the style attributes of 'headline' first to the text file, and then the style attributes for the anchor <a> tag kick in, which maintains the sizing of the 'headline' tag, while also becoming clickable as a link.

The other styles are used to highlight key terms such as the 'emphasized' class, which is bigger and bolder than the regular paragraph <p> text and is used as a text highlight.

So, the text and CSS are all ready to roll. Up to this point, you've been working blind and haven't been able to see how this all looks. It's time to change all that.

If the CSS isn't loaded and ready before anything else, the text will load with no styles attached

Open the file 'design.fla' in the 'dotnetrotator' folder. On the 'text' layer, it's possible to draw out a dynamic text field by clicking the Text tool and dragging on the stage (which is 331x242 pixels). This is where the text will be imported into. It's important that this text field is set to 'multiline' in the Property Inspector and that 'render as HTML' is selected. If this option isn't selected, the Flash file can't render the text as HTML, and the CSS and HTML formatting will not be shown.

The instance field name is 'news_txt'. This is the name the ActionScript will use to locate this text field and pass the HTML text and CSS styles into.

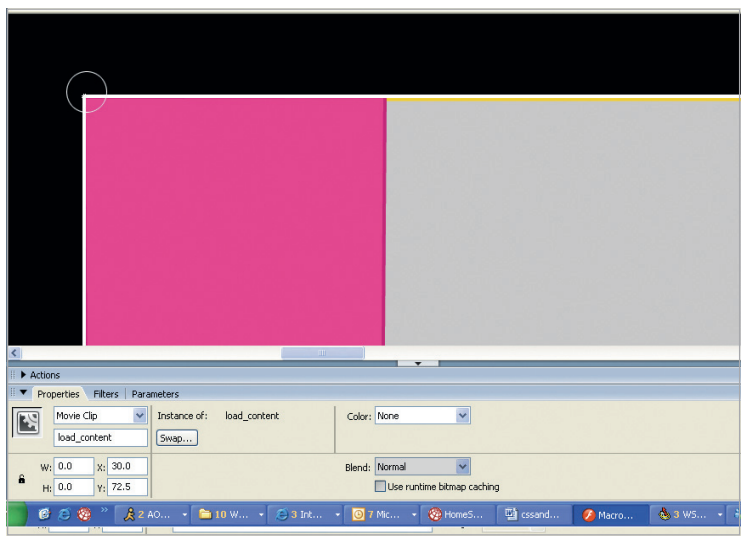
There are two other layers which handle the importing of a JPEG image. The 'preloader animation' layer holds an animation that plays progressively as the JPEG preloads. Another layer contains a blank movie clip, which this image loads into. (See 'Making a JPEG preloader animation' on the previous page.)

Loading the styles with ActionScript

The other layer in the file is the 'actions' layer. This contains the ActionScript that preloads the image and loads the text and CSS. Click this layer and scroll down until you reach 'code for loading CSS and text into swf'.

Expert tip Importing external files into Flash

The golden rule when importing files into Flash is to keep all the files relative to one another. Think about how you structure your folder hierarchy. Make sure that Flash targets the right folder and the right file. Most issues created in Preview mode are due to images, text or CSS files having been moved from their original location. Flash won't magically update the links to these files in the ActionScript, it will display a blank space. Always check this first – most of the time, it's a simple matter of putting your files back.



Blank movie clips This clip will load the external SWF files containing CSS-formatted text elements. Click on the small circle (enlarged here) to select and create an instance name

The initial part of the ActionScript loads up the CSS first. It's absolutely vital that you load in the CSS before you load in the text. If the CSS isn't loaded and ready before anything else, the text will load with no styles attached to it and will look like an unformatted mess.

```
function getRotatorStyles() {
 news_txt.text = "loading content";
 var flash_css = new TextField.StyleSheet();
 flash_css.onLoad = function(success:Boolean) {
 if (success) {
 news_txt.styleSheet = flash_css;
 getRotatorContent();
 } else {
 news_txt.htmlText = "Error - could not load CSS.";
 }
 };
 flash_css.load("css/styles.css");
 };
```

The first part of this creates a function that's going to pull in the rotator (our content box's) styles – namely 'getRotatorStyles()'.

As the Flash file begins to load, there may be a momentary lapse while Flash locates the text and the CSS. Although this may be slight, it's still important that you add a loading message to assure visitors that something is happening. Otherwise, they may simply head off elsewhere.

```
news_txt.text = "loading content"
```

Next, create a new style sheet object associated with the text field. You're informing Flash that, in effect, you're binding the two together.

```
var flash_css = new TextField.StyleSheet();
```

Now you can apply a conditional statement. This is an 'either, or' action, depending on the success of loading the style sheet.

```
if (success) {
  news_txt.styleSheet = flash_css;
  getRotatorContent();
   } else {
   news_txt.htmlText = "Error - could not load CSS.";
```

If successful, the style sheet is loaded into the text field named 'news_text', then the ActionScript moves on to load the CSS.

```
flash_css.load("css/styles.css");
```

The next step in this process is executing its other function, which will load in the text from the external file:

```
"getRotatorContent();"
```

The 'else' statement will create an error message within the text field, which lets the site visitor know that the CSS didn't load correctly. It creates a HTML text message that will display in the dynamic text field.

Loading the text with ActionScript

The styles are all loaded. What happens now? In a nutshell, the ActionScript begins to look for the text file to load.

```
function getRotatorContent() {
  news_txt.htmlText = "Loading content";

   var my_lv:LoadVars = new LoadVars();
   my_lv.onLoad = function(success){
   if (success){

   news_txt.htmlText = my_lv.info;
    } else {
   news_txt.htmlText = "A problem has been detected";
    }
    };
  my_lv.load("text/design_text.txt");
}

getRotatorStyles();
```

This is similar to loading in the CSS file. You create a new function and name it 'getRotatorContent'. There may be a lag while the content loads, so again, let the visitor know that something is happening via the message "loading content".

You need to load all the variables in the text file – the text itself – into a text field. To do this, you create a LoadVars object. This functions in a similar way to using a Movie Clip Loader (MCL) for loading JPEGs and SWFs. LoadVars loads in variables, so you need to give it something to load up. In your text file, add the following to the very start of the content:

```
&info=
```

Everything that follows from this point is now classed as a variable. As you've named it, the ActionScript will be able to find and load it into the text field. Create a LoadVars object named 'my_lv' to load the variables into. Again, you're using a conditional statement ('if') to check that the variables have loaded correctly.

```
my_lv.onLoad = function(success){
  if (success){

  news_txt.htmlText = my_lv.info;
```

If everything has gone according to plan, the code above executes correctly and the 'news_txt' text field will have the HTML formatted text loaded into it.

```
my_lv.info;
```

Here, you state that the variables to be loaded into your LoadVars object are coming in from the variable called 'info', which you placed at the start of the text content file. Again, if there's an error, an error message is generated by the line:

```
            } else {
                  news_txt.htmlText = "A problem has been detected"
```

After the conditional statement is closed, you then instruct Flash to load the text file from the specified location:

```
my_lv.load("text/discover_text.txt");

getRotatorStyles();
```

Resources Find out more online

ActionScript Developer Community
A fantastic developer's community, with forums bursting at the seams with great advice and tips. Got a question about Flash or ActionScript? Then this is your first port of call for the answers.
www.actionscript.com

Kirupa
Great support for those in need of answers to their Photoshop, PHP or ActionScript concerns. Updated regularly and easy to navigate, Kirupa is a superb resource for brushing up on your web skills.
www.kirupa.com

That's the code out of the way. Export a SWF and test the file. If nothing is showing, then make sure that the files are in the locations specified in the ActionScript. You should have a content block of nicely formatted text.

Tidying up

If some of your text is running over and has expanded below the depth of the text field, drag and drop a UI scrollbar component onto the text field. (Find this via **Window > Components > User interface**.) This will enable scrolling of the text box, and enable you to add much more content.

Open up 'rotator.fla'. This is a simple loader with three buttons that load in the content SWFs (one of which you've just produced). The size of the content placement SWFs is 586x255 pixels. In the 'Rotator' file, the key loader area is the same size. Create a blank movie clip and give it the instance name of 'load_content', then position it at X:30, Y:72.5.

The next step is referencing this instance name in the ActionScript to load in the SWFs when the three buttons are clicked on. Each button needs its own instance name ('develop_btn', 'discover_btn' and 'design_btn'). After the animated box loads, add the following code into your actions frame:

```
stop();

develop_btn.onRelease = function() {
   load_content.loadMovie("develop.swf");
   }
discover_btn.onRelease = function() {
   load_content.loadMovie("discover.swf");
   }
design_btn.onRelease = function() {
   load_content.loadMovie("design.swf");
   }
```

This creates a new function ('load_content') to load the relevant SWFs into a blank movie clip. Check the file paths if you find that the SWFs aren't loading correctly. By also adding the same code, but for just one of the SWFs, you'll find that you have a movie loading at the same time as the animation finishes loading.

Finally, export the 'Rotator' file to test it out. Check the buttons work, and make sure the right SWFs are loading for each one. Job done!

About the author

Name Paul Wyatt
Site www.paulwyatt.co.uk
Areas of expertise Website design, Flash, animation, interactive design and motion graphics
Clients 2Entertain, Lycos Europe, Smirnoff, Universal, Five
I'd most like to meet ... Keith Richards would be good for a laugh