

# Flash 8 Essentials

Paul Barnes-Hoggett  
Stephen Downs  
Glen Rhodes  
Craig Swann  
Matt Voerman  
Todd Yard



# Flash 8 Essentials

Copyright © 2006 by Paul Barnes-Hoggett, Stephen Downs, Glen Rhodes, Craig Swann,  
Matt Voerman, Todd Yard

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means,  
electronic or mechanical, including photocopying, recording, or by any information storage or retrieval  
system, without the prior written permission of the copyright owner and the publisher.

ISBN (pbk): 1-59059-532-7

Printed and bound in the United States of America

9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor,  
New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com),  
or visit [www.springeronline.com](http://www.springeronline.com).

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA  
94710. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto:info@apress.com), or visit [www.apress.com](http://www.apress.com).

The information in this book is distributed on an "as is" basis, without warranty. Although every precaution  
has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to  
any person or entity with respect to any loss or damage caused or alleged to be caused directly or  
indirectly by the information contained in this work.

The source code for this book is freely available to readers at [www.friendsofed.com](http://www.friendsofed.com) in the  
Downloads section.

## Credits

**Additional Material**      **Assistant Production Director**  
Chris Mills                      Kari Brooks-Copony

**Lead Editor**                      **Production Editor**  
Chris Mills                      Katie Stence

**Technical Reviewer**          **Compositors**  
Marco Casario                      Dina Quan and  
Van Winkle Design Group

**Editorial Board**  
Steve Anglin, Dan Appleman,  
Ewan Buckingham, Gary Cornell,  
Tony Davis, Jason Gilmore,  
Jonathan Hassell, Chris Mills,  
Dominic Shakeshaft, Jim Sumser

**Proofreader**  
Elizabeth Berry

**Indexer**  
Lucie Haskins

**Project Manager**              **Artist**  
Kylie Johnston                      April Milne

**Copy Edit Manager**          **Interior and Cover Designer**  
Nicole LeClerc                      Kurt Krames

**Copy Editors**                      **Manufacturing Director**  
Damon Larson, Julie Smith, Nicole LeClerc      Tom Debolski

## 2 BLENDING MODES



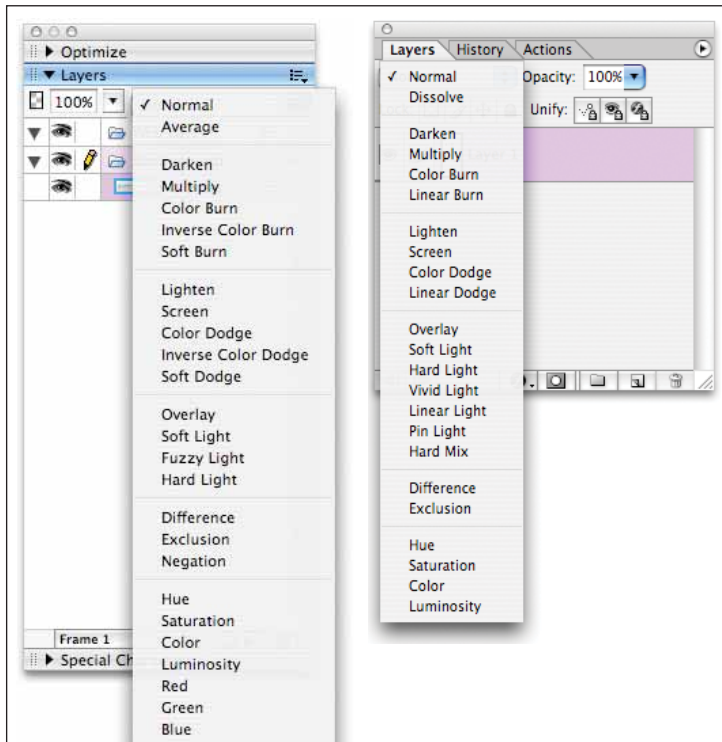
by Glen Rhodes and Craig Swann

Since the introduction of Flash as a tool for interactive development, it has widely been used as a creative tool. From online design, art, and animation, Flash has been adopted by both designers and creative people worldwide, in order to bring their visions to the world via the Internet. However, in the last few iterations of this incredible software package there was a significant movement into the world of application design and Rich Internet Application (RIA) development. As Flash grew, it became integrated and geared more towards the developer than the artist, but this is changing dramatically in Macromedia Flash Basic 8 and Macromedia Flash Professional 8!

It is clear, as will be demonstrated throughout this book, that the focus of this version is to return to the roots of visual design, for both designers and developers alike. You'll find many incredible advancements in visual display when you explore Macromedia Flash 8, including Filters, Drawing Improvements, Video, and the new Image API. But first we'll take a look at blending modes. It's time, once again, to get visual with Flash!

## So, what, exactly, are blending modes?

Blending modes are used to take pixel color values from two separate movie clips or buttons and then perform a set of calculations to create a new hybrid image. This hybrid image is a result of the calculations made on the two overlapping objects.



If you are familiar with graphic editing programs like Fireworks or Photoshop, then there is a good chance you have encountered and used blending modes before in your graphic work. Figure 2-1 is an example of the blending modes available in these applications.

**Figure 2-1.** Layer palettes in Photoshop and Fireworks

In these applications, blending modes are used between two images which are found on adjacent layers. Flash, however, has a completely different approach to stacking images, Sprites, movie clips, and objects in the authoring environment. Of course, you have the power to control your structure any way you like and you can easily drop a single instance of an image or object on its own independent layer. However, Flash also allows you the opportunity to place multiple objects on a single layer as well as the ability to attach and duplicate movie clips dynamically with code (which relies on levels and not layers). This is an important concept to understand, because with Flash, blending modes are operated on the movie clip, sprite, or button found immediately below the movie clip, button, or sprite to which the blending mode is applied.

*It is important to understand that blending modes can only be applied to movie clips or buttons. Of course, you can use static or dynamic text, video files, or live camera images, but you need to ensure that these types of content are embedded into a movie clip which you can then use to apply blending modes. Also, only the top or source of the blending needs to be a movie clip. When this movie clip is blended it will impact whatever is below it, whether it is a movie clip, text, or symbol on the stage.*

## Blending modes supported by Flash 8

Due to the complexity of some blending modes used in other popular imaging applications, such as “Hue” and “Luminosity” (which requires a color space change and thus extra processing power which does not allow for the real-time display in the Flash environment), Flash 8 uses a subset of the most common and popular blending modes.

Before getting into the nuts and bolts of each of the available blending modes and the visuals that can be produced with them, let’s quickly get an overview of the sort of blends that we can access and utilize with Flash 8.

Flash 8-supported blending modes include:

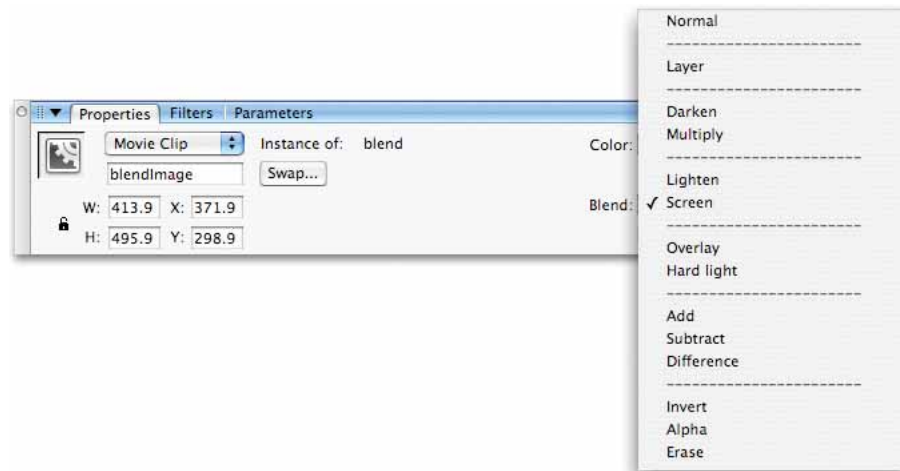
- |              |              |
|--------------|--------------|
| ■ Normal     | ■ Add        |
| ■ Darken     | ■ Subtract   |
| ■ Multiply   | ■ Difference |
| ■ Lighten    | ■ Invert     |
| ■ Screen     | ■ Layer*     |
| ■ Overlay    | ■ Alpha*     |
| ■ Hard Light | ■ Erase*     |

*\* These last three are special blending modes that you may not be familiar with and that require a different set of circumstances to use. These will be covered independently later in the chapter.*

The other great thing about the addition of blending modes in Flash 8 is the ability to access and control them using both the Flash 8 IDE and the ActionScript API. This allows you to utilize interesting visual effects, either through direct manipulation on the stage via the Property Inspector or through ActionScript by modifying a movie clip's `blendmode` property. We'll take a look at the ActionScript method in a little bit. For now, let's focus on applying blends with the Flash IDE, so that we can see exactly what these blends look like.

## Applying blends using the Flash 8 IDE

In Flash 8, the Property Inspector has been overhauled, allowing for a whole range of new visual options and parameters, and including Blends and Filters (which we will cover in the following chapter). You can now directly apply blends to movie clips with the newly added Blend drop-down menu, accessible in the Property Inspector and shown in Figure 2-2.



**Figure 2-2.** Blending modes can now be applied directly to a movie clip using the Property Inspector.

To help you understand the impact that blending modes have on images, we'll go through each of the possible modes and describe how the blending mode works, as well as demonstrate each with an example. If you are going through this book with your computer nearby, you can open the `BlendingModes.fla` and work through these examples in real-time. If, however, you are enjoying a sunny day and sitting in a park learning about the wonders of Flash 8—sigh—lucky you. You can easily follow along without a computer and see what these new blends hold in store for you.

If you are super anxious to see blending modes in action, you can skip down to the Applying blends using ActionScript section. You can also download the file `BlendingModesAS.fla` from [www.friendsofed.com](http://www.friendsofed.com), which allows you to dynamically swap blending modes to quickly see the new power in visual expression using blending.

But let's start at the beginning. Opening the `BlendingModes.fla` file will reveal a simple Timeline architecture with two layers. Two image movie clips have already been created and placed on their own layers, so you can immediately start playing with blending. You'll notice that the top layer and subsequent movie clip are labeled `sourceMC`, and the bottom layer and movie clip are correspondingly labeled `destinationMC`. Since blending modes are applied in a top-down fashion, it is the pixel information of the `sourceMC` that will be *applied* to the `destinationMC`. For those free from the chains of a computer, you can see the `sourceMC` and `destinationMC` being used in the examples shown in Figure 2-3.



**Figure 2-3.** The `sourceMC` and `destinationMC` images used for the following blending examples

### Normal mode

When accessing the Blend drop-down menu (see Figure 2-2) in the General tab of the Property Inspector the default mode will always be Normal. This mode does not mix or combine pixels of the source image with the destination image in any way. Thus, in normal blending mode, if you test the `BlendingModes.fla` file, you will generate the image shown in Figure 2-4.



**Figure 2-4.** `BlendingModes.fla` output with `sourceMC` set to Normal. No pixels are combined.

## Darken mode

This mode is used to punch through the darker colors in the source image onto the destination image. When performing the Darken blending mode, pixel colors from both sourceMC and destinationMC are compared and only the values in the sourceMC that are darker than the pixel in the destinationMC are used in the updated image.

If you change the blending mode of sourceMC in the Property Inspector to Darken you will create the new, combined image shown in Figure 2-5.



**Figure 2-5.**  
Affected image when Darken  
blending mode is applied from  
sourceMC to destinationMC

You can clearly see that the chocolate egg in the sourceMC is now visible (as is the other chocolate goodness) over the church in the destinationMC. This mode can generate an interesting result when the source image is black and white or contains high contrast. By now, even if you are not familiar with blending operations, you should start to see the new creative possibilities that exist in leveraging assets in Flash to create wonderful new compositions.

## Multiply mode

Similar to the Darken mode, Multiply mode does just what you might expect it to do. It multiplies the pixels in both the sourceMC and destinationMC. Unlike Darken, which substitutes and displays the darker of the two overlapping pixels, the Multiply mode multiplies the two color values. The resulting color will always be as dark as either of the two colors from both sourceMC and destinationMC (shown in Figure 2-6).



**Figure 2-6.**  
Affected image when Darken  
blending mode is applied from  
sourceMC to destinationMC

This image looks very similar to the last example using Darken; however, you should be able to see that some areas have become darker still as a result of the multiplying effect. Keep in mind that multiplying a color with black will always result in having black in the resulting image, while multiplying with white will always leave the destinationMC colors unchanged.

### Lighten mode

As expected, the Lighten mode will do the exact opposite of the Darken mode. Here, the lighter color of both the sourceMC and destinationMC are chosen. If the sourceMC is lighter than the pixel beneath the destinationMC, this color is transferred to the destinationMC in the resulting image—otherwise it is left unchanged (shown in Figure 2-7).



**Figure 2-7.**  
Affected image when Lighten  
blending mode is applied from  
sourceMC to destinationMC

In Figure 2-7 you can clearly see that only the lighter elements from the sourceMC (in this case, the icing and feather) are displayed over the building. This type of effect is often used in superimposing text and titles over images and video, in order to create more fluid and soft-edged type treatments.

## Screen mode

Just as the Lighten blending mode is the opposite of the Darken blending mode, so is the Screen mode opposite to the previously demonstrated Multiply mode. Thus, screening a color with white will produce white (as black did with Multiply) and screening with black will leave the color unchanged. Technically, the colors from both sourceMC and destinationMC are complemented and then multiplied before the destination image is replaced with the resulting image. A picture tells a thousand words, so take a look at the resulting image, shown in Figure 2-8, when Screen mode is applied to the sourceMC.



**Figure 2-8.**  
An example of an image when Screen blending mode is applied from sourceMC to destinationMC

Although it's similar to the Lighten blending mode, you can see how the images take on more of an ethereal and soft look when this blending mode is applied. This is an excellent blending mode to use when you need to create highlights or apply a lens flare dynamically to an image.

## Overlay mode

The Overlay blending mode either multiplies or screens the colors, depending on the destination color you've chosen. The effect is that the sourceMC will be overlaid over the destinationMC while maintaining all of its highlights and shadows. The resulting composition generally will contain more of the destinationMC image, as shown in Figure 2-9.

This effect looks similar to having both images on transparency. The color values will be overlaid and create a blend of colors between source and destination. However, unlike transparency, highlights from the source image will remain in the resulting image.



**Figure 2-9.**  
An example of an image when Overlay blending mode is applied from sourceMC to destinationMC

This mode is often used in digital cartooning and coloring. Artists will often take the outlined illustration, apply the coloring on another layer, and use the Overlay mode to create the two separate layers. It is an interesting way to combine pen-sketch illustrations and digital coloring techniques.

### Hard Light mode

Hard Light blending mode operates very similarly to the previous Overlay mode, with the exception that color values from sourceMC are used instead of destinationMC for determining whether overlapping pixels are screened or multiplied. For instance, if the sourceMC color is lighter than 0.5, the destinationMC is lightened through screening. If the sourceMC color is darker than 0.5 then the destinationMC is darkened through multiplying. Subsequently using pure black or pure white will maintain color values as shown in Figure 2-10.



**Figure 2-10.**  
Example of image when Hard Light blending mode is applied from sourceMC to destinationMC

Just as Overlay generally maintained more of the destinationMC image, you can see from Figure 2-10 that the sourceMC becomes more prevalent when you're using the Hard Light blending mode.

## Add mode

The Add blending mode adds the sourceMC to the destinationMC color values to create the resulting image. The general result is a soft, bright image, which can be useful when creating dissolves between images as well as for accomplishing lighting type effects (see Figure 2-11).



**Figure 2-11.**  
Affected image when Add  
blending mode is applied from  
sourceMC to destinationMC

## Subtract mode

With the Subtract blending mode, the reverse of the Add blending mode is calculated and the sourceMC is subtracted to the destinationMC. This blending mode can be used for shadow-type effects, as shown in Figure 2-12.



**Figure 2-12.**  
Affected image when Subtract  
blending mode is applied from  
sourceMC to destinationMC

## Difference mode

This unique blending mode often creates surprising results. The Difference blending mode operates by determining the darker of the two colors in sourceMC and destinationMC and then subtracting the darker of the two from the lighter one. This way, white will always invert the destination color and black in the sourceMC will create no change. Generally this effect will create the vibrant saturated colors shown in Figure 2-13.



**Figure 2-13.**  
An example of an image when Difference blending mode is applied from sourceMC to destinationMC

## Invert Mode

Invert mode is a common image manipulation in many graphic applications and involves inverting the colors of the destinationMC image. However, you can't apply this blending mode to a single destinationMC. Here, the sourceMC is used merely to represent the area of the destinationMC that you wish to invert. Similar to a mask, only the overlapping areas will invert the destinationMC (see Figure 2-14).



**Figure 2-14.**  
An example of an image when Invert blending mode is applied from sourceMC to destinationMC

It's important to know that this blending mode is affected differently with regards to the alpha of the sourceMC. The strength of the sourceMC determines the level of invert on the destinationMC. Applying an alpha level of 100 to the sourceMC creates a full invert; however if you try adjusting the alpha of the sourceMC to 50% you will notice that the entire image goes grey, and that reducing it further will reduce the invert to 0%, which will create no invert whatsoever. This is a significant difference from the use of the mask and you should be aware of this if you plan to animate MovieClips with alpha while using an invert blending mode.

Now that you are familiar with what will ultimately be the most familiar blending modes available in Flash 8, it's time to take a look at several more blending modes, all of which operate a little differently than the modes covered so far. To access the Alpha, Erase, and Layer blending modes we need to create a Composition clip to properly achieve the blending effects.

## Applying Layer, Alpha, and Erase blending modes

With the addition of these new blending modes comes a special set of operations that have been long awaited. For years, creating a soft mask in Flash has been no easy task. Creating mask layers with alpha has never produced the soft-faded mask that you might have expected—instead it has always assumed the shape of the object and not the alpha information it contains. However, many wishes have been fulfilled with the addition of the new Layer blending mode, which finally allows you to create soft-feathered masks using alpha gradients.

Using the Layer blending mode, however, involves a few more steps than the blending modes covered so far, but the payoff is well worth it. So, let's get on with it.

### Layer blending mode

As mentioned, using the Layer blending mode requires some additional steps to generate the desired effects, and uses both Alpha and Erase blending modes. These sets of blending modes can work in conjunction, but they require the creation of an additional MovieClip for compositing the final result.

The reason this extra composition movie clip is required is because unlike an imaging application that uses a layer hierarchy, such as Fireworks or Photoshop, Flash uses an entirely different hierarchical structure for managing elements and objects on the stage. For this reason, the use of Alpha and Erase require that an additional parent movie clip is set to the Layer blending mode. Flash treats this movie clip as a new canvas where embedded blending modes are calculated and then parent clips are redrawn using normal mode. This process is necessary because Flash can't modify the opacity of the main or root timeline and both of these blending modes use opacity and thus alpha modifiers.

But enough with the technical explanations—no doubt you are anxious to see how easily we can create soft masks!

## Alpha mode: Creating soft masks

The purpose of the Alpha blending mode is to utilize the alpha information of the applied movie clip inside the composition Layer movie clip in displaying the destinationMC image. Any area in the movie clip that is transparent (using alpha values) will cause the same area in the destinationMC to be transparent, allowing the sourceMC image to show through. This may seem confusing at first, so let's take a look at the following example which clearly illustrates the proper structure required to use the Alpha blending mode. When completed, you will have a feathered circle mask displaying the destinationMC through the sourceMC.

Start by opening `BlendingModes.fla`, which is the file you used in the previous examples. You will be modifying this file yourself, in order to familiarize yourself with the concept of using Layer and Alpha blending modes. If you want to jump right ahead and see the finished product, you can take a look at the `AlphaBlending.fla` which is the completed file created by following the steps below.

### Creating a composition movie clip using the Layer blending mode

The first thing that you need to do is modify the existing sourceMC blend and transform it into a precomposition movie clip.

1. Select sourceMC, which is located on the top layer (you may want to lock the destinationMC layer, as you won't need to modify it in this example) and change the blending mode to Layer using the Blend drop-down menu on the Property Inspector.

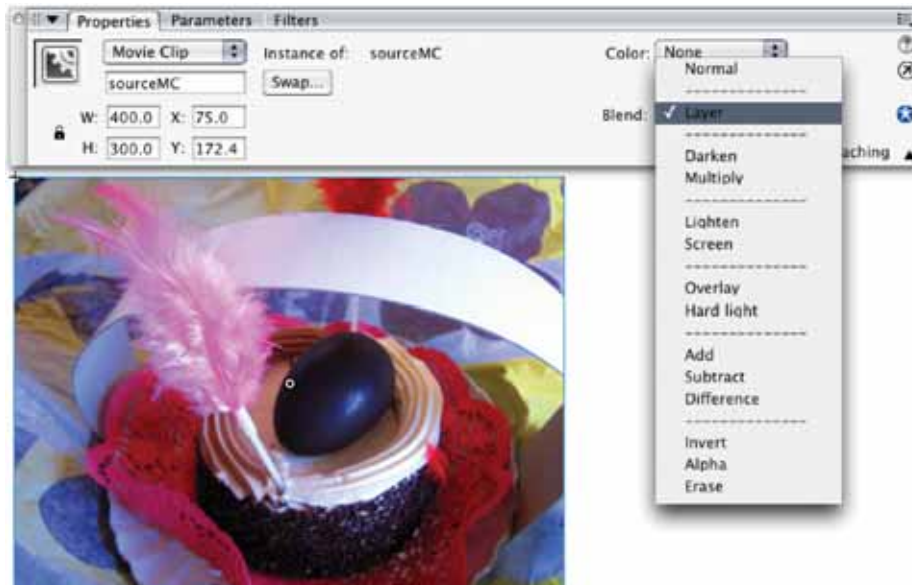
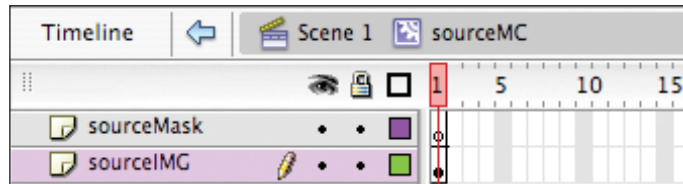


Figure 2-15. Setting the parent composition movie clip to the Layer blending mode

## FLASH 8 ESSENTIALS

You'll notice that it appears the same as it would in the Normal blending mode. Your sourceMC image is the only thing visible blocking out the destinationMC image that lies beneath it.

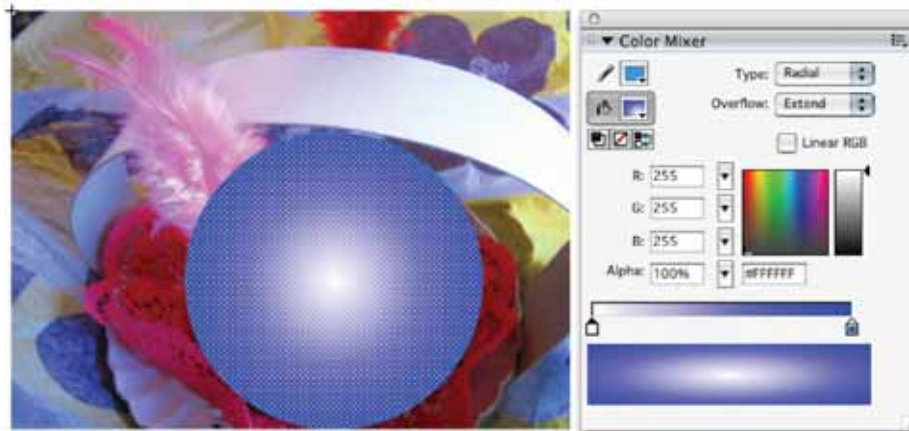
2. With sourceMC set to the Layer blending mode, double-click on sourceMC to enter "Edit in Place" mode, and add a new layer on top and label it sourceMask.



**Figure 2-16.** Creating a layer for Alpha blending mode to be applied to sourceMC Timeline

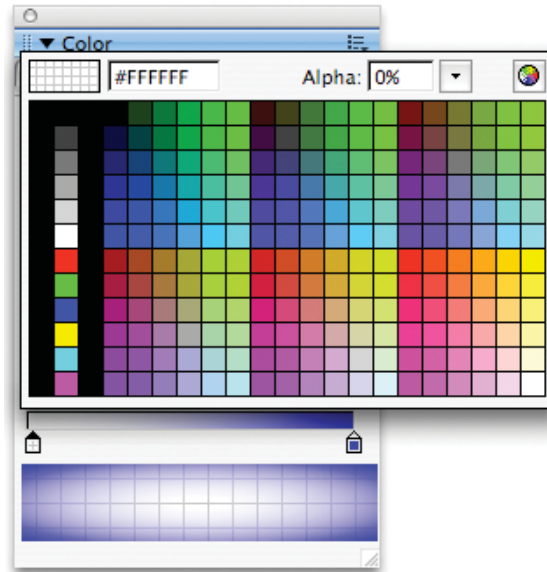
The sourceMask layer is the layer that will contain the masking movie clip you will create next.

3. With the sourceMask layer selected, draw a circle on the stage. Look for the Color Mixer palette. If you can't find it, open the Color Mixer palette using *SHIFT+F9* and you'll find it at the right side of the screen. With the newly drawn circle selected, use the Color Mixer palette and set the fill to Radial.



**Figure 2-17.** Creating Alpha gradient to be used with Alpha blending mode

4. Since the goal is to create a soft feathered mask, you need to adjust the inner gradient alpha value and set it to 0. By doing this, the gradient fades from full opacity on the outside to a transparent center region. Select the left gradient color and modify its alpha setting to 0.



**Figure 2-18.** Setting inner gradient alpha level to 0% for displaying destinationMC

Your new gradient circle should now have an alpha fade and provide a preview of the final effect.



**Figure 2-19.** Preview of feathered mask region inside the sourceMC

However, this preview of the alpha mask is currently over your sourceMC image, which is not what we want. Remember, the object is to mask the destinationMC image through this newly created mask. The next steps are crucial for achieving this.

5. Since blending modes can only be applied to movie clips you must now select your newly created alpha gradient circle and turn it into a movie clip. Once selected choose Modify > Convert to Symbol (F8) and give it the name sourceMask.
6. To utilize this alpha information, you must now select your newly created sourceMask movie clip and change its blending mode to Alpha in the Property Inspector. Once you have done this, you'll notice that sourceMask is now invisible on the stage, with only its bounding box showing.



**Figure 2-20.**  
How the sourceMask appears inside sourceMC when set to Alpha blending mode

Fear not! Although the destinationMC clip appears invisible inside of the sourceMC movie clip, if you return back to the main Timeline by selecting the Scene1 link, you'll see that the same destinationMC clip is now being displayed through the alpha gradient clip that you created.



**Figure 2-21.**  
Final completed Alpha blend effect when previewed on main Timeline

It's as easy as that! This is just the beginning of the interesting and creative things that can be done with this fabulous new blending mode. For instance, with this file, try creating

different sourceMask clips containing different shapes, sizes, and alpha gradients. Experiment! Explore!

Another beautiful aspect of utilizing the Alpha blending mode is that you can animate them as well. For instance, in your sourceMC Timeline, try animating your newly created sourceMask clip. Try scaling or moving it and note how the alpha information is transferred to the destinationMC when you test your movie. With luck, this will be the beginning of a whole new world of creative possibilities for your Flash projects.

### Erase mode

With the Erase blending mode, you use the same process that you used with the Alpha mode—which, as you’ll remember, requires the creation of a parent composition MovieClip set to a Layer blending mode.

The generated effect, however, is the opposite of what was seen with the Alpha blending mode. When you use Erase, the opaque areas of the alpha gradient remove areas of the destinationMC. Areas with low alpha will allow the destinationMC to be seen through the sourceMC. You can see this opposite effect in the following image.



**Figure 2-22.**  
Final completed Erase blend effect when previewed on main Timeline

Although the blending modes demonstrated so far in this chapter have been done manually in the Flash IDE, blending modes are also available as a `MovieClip` property and can be accessed via `ActionScript`—and it couldn’t be easier!

### Applying blends using `ActionScript`

In Flash 8, the ability to access blending modes has been extended beyond the IDE and is now accessible through `ActionScript`. You can now modify a movie clip’s blending mode by altering the new `blendMode` property. Macromedia defines this new property by saying

## FLASH 8 ESSENTIALS

that `blendMode` is “The blending mode for this movie clip. The blending mode affects the appearance of the movie clip when it is in a layer above another object on-screen.”

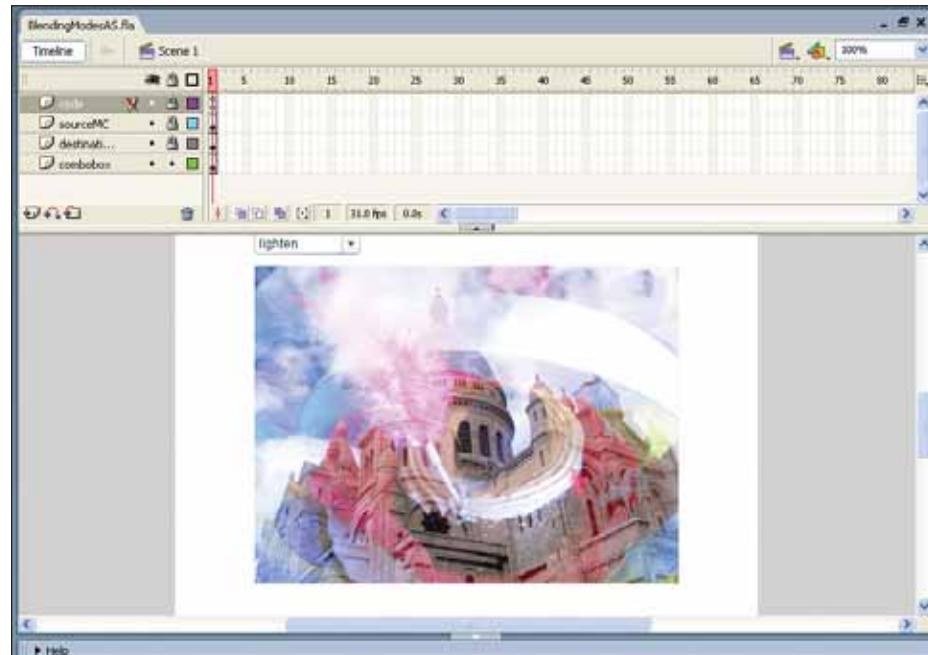
1. Reopen the `BlendingModes.fla`, if you don't still have it open.
2. Create a new layer and label it `Actions`
3. On frame 1, create a frame action with the following code:

```
sourceMC.blendMode = "darken";
```

If you test this movie now, you will see that this code will override whatever existing blending mode was set manually in the Property Inspector. The `blendMode` property accepts both strings and numbers (1–14) as values, so if you are using a string, ensure that you have it within quotations. For instance for the `Multiply` blend you can either use:

```
sourceMC.blendMode = "multiply";  
sourceMC.blendMode = 3;
```

If you'd like to be able to see how all the blending modes look quickly, open up the `BlendingModesAS.fla` file. This file is the same as `BlendingModes.fla` with the addition of a `ComboBox` component which dynamically will change the blending mode of the `sourceMC` movie clip when a new selection is made from the `ComboBox`.



**Figure 2-23.** `BlendingModeAS.fla`, ready to go.

The additional code used to perform this is also very simple as you can see:

```
modeChangeListener = new Object(); // Create the listener object
// define the 'change' event
modeChangeListener.change = function(evtObj)
{
    // Set the blendMode to the value of the label
    //of the selected item in the blend_cb comboBox
    sourceMC.blendMode = evtObj.target.selectedItem.label;
}
// Assign the change event to blend_cb
blend_cb.addEventListener("change", modeChangeListener);
```

The ComboBox component is merely passing on the label which is a string to the blendMode property in the function called when a selection is made in the ComboBox.

## Summary

And so concludes today's lesson on the incredible and amazing blending modes now supported in Flash 8. However, this is just the tip of the iceberg, with regards to the new visual enhancements available within Flash 8, as you'll soon see. In the next chapter you will discover the wonderful new world of Flash filters.

**Table 2-1.** The summary of the different blending modes

blendMode	Result of sourcePixel and destPixel
Normal	sourcePixel
Darken	if (sourcePixel < destPixel) sourcePixel else destPixel
Multiply	sourcePixel x destPixel
Lighten	if (sourcePixel > destPixel) sourcePixel else destPixel
Screen	$1 - (1 - \text{sourcePixel}) \times (1 - \text{destPixel})$
Overlay	if (sourcePixel < 0.5) Multiply sourcePixel with destPixel Else Screen sourcePixel with destPixel

*Continued*

**Table 2-1.** The summary of the different blending modes (*continued*)

<b>blendMode</b>	<b>Result of sourcePixel and destPixel</b>
Hard Light	<pre> if (destPixel &lt; 0.5)   Multiply sourcePixel with destPixel Else   Screen sourcePixel with destPixel </pre>
Add	<code>sourcePixel + destPixel</code>
Subtract	<code>(sourcePixel + destPixel) - 1</code>
Difference	<code>  sourcePixel - destPixel  </code>
Invert	<code>1 - destPixel</code>

Note that, in all cases, the math is performed on the red, green, and blue component values separately, to create final pixel red, green, and blue components. So, Multiply actually does the following:

```

newRed = sourceRed x destRed
newGreen = sourceGreen x destGreen
newBlue = sourceBlue x destBlue
final pixel is newRed, newGreen, newBlue

```

Though values are represented as a number from 0 to 255 digitally, they are actually treated as if they are in the range of 0 to 1 when the blend operations take place. This way, multiplying  $128 \times 128$ , which would normally be 16,384, would actually be treated as  $0.5 \times 0.5$ , which would result in 0.25, and then digitally that would be 64.

In some cases, for example with Add and Subtract, where a resulting value is greater than 1 (255) or less than 0, then the value is simply clamped and set to 255 or 0.