

Working with Photos

Flash allows you to *import* various types of images, and it distinguishes between two types: vector and bitmap. Photographs are always bitmaps. An image you create within Flash is always a vector image, whether it is a simple rectangle or a complex diagram. Both bitmaps and vectors can be imported from outside Flash.

- A *vector* image can be scaled (made larger or smaller) with absolutely no difference in quality: The lines remain smooth and the solid colors remain solid. The SWF file size does not increase significantly if you use gigantic vector images instead of small ones.
- When a *bitmap* image is scaled larger, the quality decreases: The edges become jagged, and the individual pixels become larger, making the entire image appear blocky and unnatural. Bitmaps with large file sizes will increase the final SWF file size significantly.

The file types JPG, GIF and PNG are all bitmaps when imported into Flash, as well as the file types BMP and PICT.¹ (You may be familiar with these file types from previous Web or photo work you have done.) Bitmaps are always created and edited in programs outside Flash.

You can import vector images to Flash from some other programs that create vector images; these include Macromedia's Fireworks and Freehand and Adobe's Illustrator.

Now that you know the basic terminology, just remember that Flash uses the term "bitmap" very broadly—and the term includes *more* than photographs. Photographs are *always* bitmaps in Flash, no matter what file format they have when you import them. Photojournalists typically save their photo files in the JPG format, but photos can be saved in other formats too.

About This Lesson

This lesson explains tasks that require you to use some features and properties of Flash that novice users don't necessarily need or know much about. These are tasks of particular interest to photographers and photojournalists. After you have worked through all the exercises, you should be able to do just about anything you envision doing with photos in Flash.

This lesson includes several exercises and informational sidebars. You may not want to work through each one of the exercises at this time, but note that the first two exercises, about importing and optimizing photo files, contain information that can make *a very*

¹ For a complete list of the file formats that can be imported into Flash, see the Flash Help files: Using Imported Artwork and Video > Placing artwork into Flash > Import file formats for vector or bitmap files. The acceptable file formats normally include JPG, GIF, PNG, BMP, and PICT. If QuickTime 4 or later is installed on your system, you can also import PSD and TIFF files.

significant difference in the final file size (and download time) of your Flash movies. You may want to simply *read* those two exercises rather than work through them, but it's important not to skip them.

Depending on your reasons for using a photo in Flash, you may or may not be concerned about having the highest possible image quality. For example, a background image with text over top of it will not need to be sharp and perfect—so you could select a lower quality setting and save a lot of kilobytes in file size. Whether you want your photos to look perfect or not, this lesson will show you how to get the results you want in the final SWF.

In This Lesson

- Sidebar: Photo Editing Tips
- Exercise: Importing and Optimizing Compressed Photo Files
- Exercise: Importing and Optimizing Uncompressed Photo Files
- Sidebar: Bitmap Properties: Photo vs. Lossless
- Exercise: Moving Photos in Flash
- Sidebar: Motion Tweens, Bitmaps, and Symbols
- Exercise: Zooming In on a Photo
- Exercise: Fading Photos into Each Other
- Exercise: Looping a Fade Effect
- Sidebar: Doing It All with Movie Clips Instead
- Exercise: Using a Photo as the Background
- Sidebar: Importing a Sequence of Images

Lesson 7: Working with Photos

Before you begin this lesson, make sure you have at least *two versions* of the *same* photo file. Edit the photograph in Photoshop or a similar program. Crop or resize the photo to a *width* of 500 pixels and a *height* of 300 pixels. (If the photo is vertical rather than horizontal, make sure the height is *no more than* 300 pixels.)

For the first version, save the photo in the JPG file format, with a quality of medium or high; this is the *compressed* version of the file. For the second version, save the same photo as a BMP (Windows) or PICT (Mac); this is the *uncompressed* version of the file.

Sidebar: Photo Editing Tips

1. Decide on the screen size (width and height) of your Flash movie *before* you edit your photos. (See Lesson 1, Step 3, if you need instructions.) That way, you'll know the maximum width and height for your photos.
2. Because more than 40 percent of Internet users still use a screen resolution of 800 x 600 pixels,² the best practice is to set the width and height of your Flash movie to accommodate that resolution. The largest dimensions considered acceptable: 700 pixels (width) by 400 pixels (height). This size allows the SWF to be displayed in the browser window without requiring the user to scroll. (The buttons at the top of the

² Data from May 2003; see— <http://www.thecounter.com/stats/2003/May/res.php>

browser window take up a lot of space, which is why 400 pixels is the maximum height for the Flash movie.) If you're using a pop-up window without browser buttons, menu bar, scrollbars, or status bar, you can go to 750 x 450.

3. You cannot efficiently edit a photo in Flash (although you can launch an external program and edit a photo after it has been imported to Flash). You should make sure the photo looks as you want it to look *before* you import it. That includes the *exact width and height* you want the photo to be within the Flash movie.
4. If you save your photos at a larger width and height than necessary in the Flash movie, you are increasing the SWF file size (and the user's waiting time during the download). This is not a good practice!
5. Decide whether you will put text or other elements (such as buttons) *above*, *below* or *beside* your photos. For example, if your Flash movie is 400 pixels high and your photos are also 400 pixels high, there will be no space left for anything above or below the photo. Also consider whether some text will be longer than one line.
6. Flash will not display progressive JPGs. ("Progressive" is an option you can select when you save an image as a JPG file in Photoshop and similar programs.) If you want to use a progressive JPG in Flash, you must first *resave it* in another program (e.g. Photoshop) as a non-progressive JPG.

Importing and Optimizing Compressed Photo Files

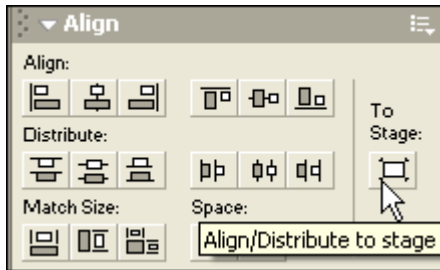
1. File menu > New (Start with a new, empty file.)
2. Change the size of the Flash movie to 600 pixels (width) by 400 pixels (height). Change the background color if you want it to be a color other than white. See Lesson 1, Step 3, if you need instructions for how to do this.
3. Open the File menu and select "Import to Library ..." A dialog box opens, allowing you to find (on your computer) the photo file you want to import to Flash. Find the *compressed* (JPG) photo file you prepared, select it, and click Open.

Note: If you want to import *more than one* photo, you can Ctrl-click (Windows) or Cmd-click (Mac) each photo filename to select several files and import them all at one time. You can Shift-click to select a contiguous set of files.

4. Open the **Library** (press F11). You will see the imported photo file there. Click once on the photo filename to see a thumbnail image of your photo in the window at the top of the Library panel.
5. To place the photo on the **Stage**, *drag it* from the **Library** to the Stage.

Note: You can import a photo directly to the Stage by selecting "Import" instead of "Import to Library ..." from the File menu (Step 3 above). Alternatively, you can copy a photo and paste it on the Stage. In professional practice, however, you are more likely to import a set of photos to the **Library** at one time and place them on the Stage afterward. No matter which method you use to import photos to Flash, all photos will appear in the Library.

- To center the photo easily on the Stage, open the **Align** panel (from the Window menu). Make sure the photo is selected (click it once). In the **Align** panel, first click the Align/Distribute to Stage button (labeled “To Stage”). Then click the second button in the top row of the panel (Align horizontal center). Then click the fifth button in the top row of the panel (Align vertical center). These two buttons allow you to simply position the photo in the exact center of the Stage.



The **Align** panel provides one-click options for aligning objects. If you want to align objects in relation to the entire **Stage**, click the button as shown at left. If you want to align objects relative to one another, unclick that button first.

When the button appears “pushed in,” all the alignments made will be relative to the entire Stage.

- Save your movie as “phototest1 fla” and test it (Ctrl-Enter or Cmd-Return). Your photo should look good!
- You don’t want your final Flash file to be so huge that users refuse to wait for it to download—and that is a very real concern when you put photos into your movie. One solution is to keep all your photos *outside* the SWF and load them externally. You will learn how to do that in Lesson 10. Here, we need to look at how to optimize photo files that are *inside* the Flash movie, because sometimes you have good reasons for the photos to be internal.

In this case, you have a *compressed* (JPG) file in your movie. Take a look at the original file outside Flash and see how large the file is (in kilobytes, or KB). For example:

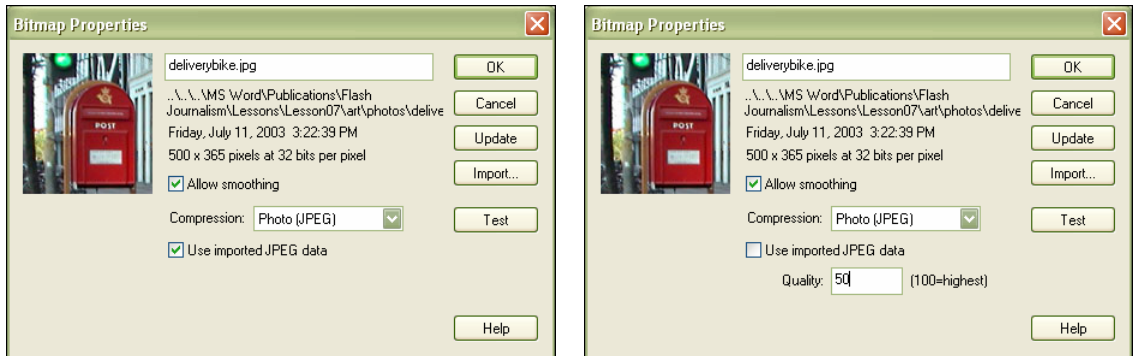
Filename	Width x Height (in pixels)	File Size (outside Flash)	JPG Quality (in Photoshop)
deliverybike.jpg	500 x 365	64.6 KB	High (60)

Then look at the file size of the SWF file you just tested (in Step 7). For the file containing the photo *deliverybike.jpg*, the SWF file size is 64.7 KB. Since there is nothing in the FLA except the photo (in one frame), you can infer the direct relationship between the file size of the photo and the file size of the SWF.

- One way to reduce the file size of the SWF, of course, is to make the photo file size smaller *before* you import it to Flash. Alternatively, you can change the amount of compression that Flash applies to the photo. It is *very important* to understand that the methods for doing this are different for *compressed* and *uncompressed* files, so keep in mind that you are working with a *compressed* file (JPG) in this exercise! (Later in this lesson you will learn the procedure for an uncompressed file and compare the results.)

In the same file you saved above, right-click (Control-click/Mac) the photo file in the **Library**. Select “Properties ...” from the pop-up menu. The Bitmap Properties dialog

box opens. *Uncheck* the box labeled “Use imported JPEG data.” Doing that allows you to change the Quality setting for this photo. *Type* “50” (without the quotes) in the Quality field, then click OK.



The default setting for JPG file compression in Flash. This keeps both the quality and the file size of the imported photo (in JPG format) the same as it was outside Flash.

Change the amount of compression applied to this single JPG image. A higher number means better quality and a larger file size. A lower number means worse quality and a smaller file size.

Note: If you check “Allow smoothing,” Flash will use anti-aliasing to smooth the *outside edges* of the bitmap. If the edges are perfectly straight, as with most photos, this has no effect, either checked or unchecked. It also has no effect on the file size in either case when the photos have straight edges.

10. Save your movie with a *new* filename (such as “phototest2 fla”) and test it (Ctrl-Enter or Cmd-Return). Your photo will not look as good as the first version of this file; it will be blurrier, more pixilated, or both. You can open the two SWFs side by side to compare the quality of your photo in each version. (The difference between the two photos may be subtle if the original was not of good quality.)

Look at the file size of the second SWF. For the file containing the photo *deliverybike.jpg* (discussed above) the file size of the second SWF is 29.9 KB, compared with 64.7 KB for the first SWF (the file with no additional compression applied by Flash). You can see that reducing the quality to 50 percent in the Bitmap Properties dialog makes a very significant difference.

Filename	JPG File Size (outside Flash)	SWF File Size (default Bitmap Properties)	SWF File Size (Bitmap Properties Quality = 50%)
deliverybike.jpg	64.6 KB	64.7 KB	29.9 KB

In many cases, you will not want to sacrifice so much image quality. However, just changing the Quality setting to 90 percent in the Bitmap Properties dialog for each imported JPG can greatly reduce the SWF file size for a file containing *several* photos!

Obviously, there’s a tradeoff to consider. You would like to make the file size of the SWF as small as possible so that it downloads quickly. However, in most cases you would not want to publish blurry or pixilated photos.

Note: The *only way* to change the quality of imported JPGs *within Flash* is to change the Quality setting in the Bitmap Properties dialog for each *individual* JPG file in the **Library** of the FLA file—as you have just done. A common misconception is that you can change the quality of imported JPGs in the Publish Settings dialog (explained in Lesson 3 and in the following exercise), but that is not true. You can use the Publish Settings dialog to change the quality and compression of *uncompressed* image files, such as BMP and PICT files, but *not* JPG files.

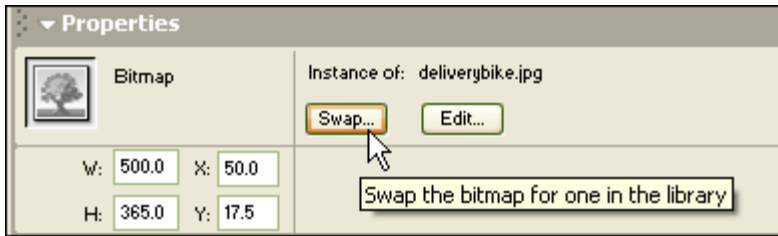
If you want to prove it, conduct a test similar to the one above: With one JPG image on the Stage in a one-frame movie, save an FLA two ways: (1) use the default JPEG Quality setting in the Publish Settings dialog, and (2) use a much smaller number in the JPEG Quality setting in the Publish Settings dialog. This number has no effect on *either* the quality of the imported JPGs, *or* on the file size of an SWF that contains only *compressed* imported images.

Importing and Optimizing Uncompressed Photo Files

In this exercise, you will do almost exactly the same thing you did above, but you will use an *uncompressed* file (a BMP or a PICT) instead of a *compressed* file (JPG).

There is a very good reason for this exercise: You may find that you can control the *quality* of imported photos better if you import *uncompressed* files and adjust the compression settings within Flash. This will be quite important to photojournalists, although it may seem less important to other people.

1. Open your file named “phototest1.flas” from the exercise above, and immediately save it as “phototest3.flas” (to allow a comparison later).
2. Open the File menu and select “Import to Library ...” A dialog box opens, allowing you to find the photo file you want to import to Flash. Find the *uncompressed* (BMP or PICT) photo file, select it, and click Open.
3. Open the **Library** (press F11). You will see *both* photo files there—your JPG from the previous exercise *and* your freshly imported uncompressed file.
4. Select the photo on the **Stage**. (That image should still be the JPG from the previous exercise.) As always, *click only once* to select an object on the Stage.
5. With the photo selected on the **Stage**, look in the **Properties** panel and find the Swap button. Click it. In the Swap Bitmap dialog box, you can *select the other file* in your **Library**—the one you just imported. (This feature is very convenient, because it places the new image in exactly the same position as the one it replaces. No need to center the new one, as long as the two photos are the same width and height.)



Open the Swap Bitmap dialog, which allows you to replace one image with another.

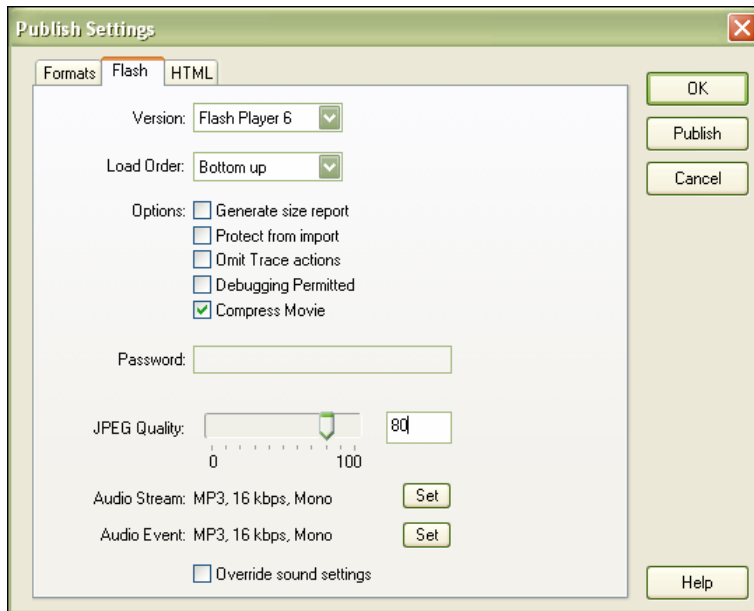
6. You can change the compression and quality of this uncompressed image in exactly the same way you did in the previous exercise, and *in another way* as well. Before you try it, check the file size of the uncompressed file outside Flash and see how large it is (in kilobytes, or KB). For example:

Filename	Width x Height (in pixels)	File Size (outside Flash)	Depth (in Photoshop)
deliverybike.bmp	500 x 365	534 KB	24 bit

This is typical—an uncompressed photo file will always be *much* larger than the compressed (JPG) file.

You might expect the resulting SWF file to be just as large as this image file, because that is what you saw in the previous exercise. To see whether that’s how Flash really works, follow the instructions in the next two steps.

7. In the **Library**, right-click (Control-click/Mac) the filename of the *uncompressed* photo. Select “Properties ...” from the pop-up menu. The Bitmap Properties dialog box opens. Make sure the box labeled “Use imported JPEG data” is *checked*. Then click OK. (You *could* change the settings here, but you’re about to see a different method for changing image compression and quality.)
8. Open the File menu and select “Publish Settings ...” In the Publish Settings dialog, click the tab labeled “Flash.” In the box labeled “JPEG Quality,” type “80” (without the quotes). Click OK.



Changes made in the Publish Settings dialog affect all uncompressed images in the FLA file (unless the individual Bitmap Properties for an image have been changed through the Library). Changes made in the Publish Settings dialog do not affect compressed images (JPGs).

9. Save and test your movie (Ctrl-Enter or Cmd-Return). Then check the file size. For example:

Filename	BMP File Size (outside Flash)	SWF File Size (JPEG Quality = 80%)
deliverybike.bmp	534 KB	53 KB

If you open “phototest1.swf” and “phototest3.swf” side by side and compare the two, you may decide that the photo quality in “phototest3.swf” is just as good as, or even slightly better than, the photo quality in “phototest1.swf”—and yet the file size of “phototest3.swf” is likely to be smaller (your results may vary, depending on your photo).

Note that if you import only *uncompressed* images (no JPGs) to your FLA file, when you change the JPEG Quality setting in the Publish Settings dialog (as you just did in Step 8), it will change the quality of *all* the imported images in the movie. That can be rather convenient when you have a lot of imported images in your Flash movie!

You can override the JPEG Quality setting in the Publish Settings dialog for any single uncompressed image by opening the Bitmap Properties dialog (right-click/Windows or Control-click/Mac on the image filename in the **Library**).

Don’t forget: The JPEG Quality setting in the Publish Settings dialog never has any effect on JPG images imported to your movie. This setting effects only *uncompressed* image files you have imported to the FLA file.

File Type	Change Quality for One	Change Quality for All
JPG (compressed)	<ol style="list-style-type: none"> 1. Right-click filename in Library 2. Select "Properties ..." 3. In Bitmap Properties, uncheck "Use document default quality" 4. Change number in Quality field 	Cannot be done.
BMP or PICT (compressed)	Same steps as above.	<ol style="list-style-type: none"> 1. File menu > Publish Settings 2. Flash tab 3. JPEG Quality: Change number in this field, or use the slider. <p>Note: If Bitmap Properties were changed (see left), those settings override these for just that one image.</p>

At a glance: Changing the quality settings for imported images.

Sidebar: Bitmap Properties: Photo vs. Lossless

In the Bitmap Properties dialog, for compression type you can choose either "Photo (JPEG)" or "Lossless (PNG/GIF)." The logic behind your selection should follow the standard practice in Web page design: Any image with complex tones or shading should be saved as a JPG (all photos and also complex illustrations with realistic detail); any image with mostly flat color should be saved as a GIF (cartoons, simple line drawings with no shading, most text and buttons).

When in doubt, save the original *uncompressed* file twice (using Photoshop and its "Save for Web" option, for example)—once as a JPG and once as a GIF—and compare the image quality *and* the file size of each one.

The PNG format is not widely supported by today's Web browsers, so it is rarely used on Web pages. Flash *does* support PNG files, so you might also experiment with the results of saving as a PNG (which is an option in Photoshop's "Save for Web").

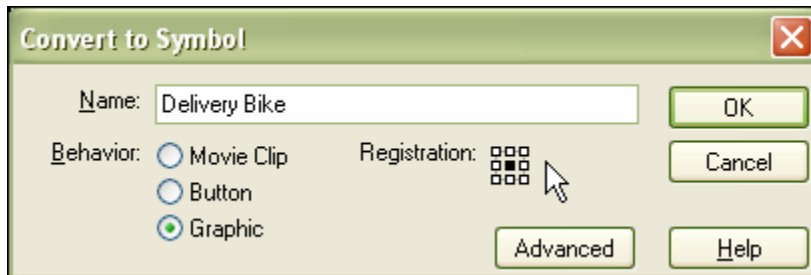
Moving Photos in Flash

Now that you've learned how to get photos into Flash, you can animate them easily. This widely used technique can be mistaken for video. The next time you see a moving photographic image on a Web site, think about whether you could achieve the same effect with one or more still photos. The answer is often yes!

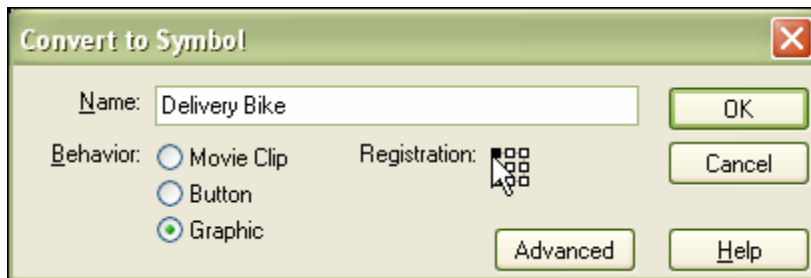
For this exercise, you can use the same file you saved in the previous exercise ("phototest3.swf"). One imported bitmap image is enough to complete this exercise. The photo you use in this exercise should be at least 400 pixels wide or high.

1. Begin with a one-frame **Timeline** and one photograph centered on the **Stage**, as in the previous exercise. (The imported photo should be an uncompressed BMP or a PICT.) Set the *frame rate* to 15 fps, to make it easy to follow the instructions in this exercise.

2. Select the photo on the **Stage** (click it once) and *convert it to a graphic symbol* (press F8). This step is absolutely crucial, because any object you move in Flash *must* be a symbol. While you have the Convert to Symbol dialog open, make sure the *registration point* for this object is in the *upper left corner*. (The reason for this will become clear very soon!) If you need to change the registration point, you must do it now: Click the box in the *upper left corner* of the 3x3 matrix, as shown below.

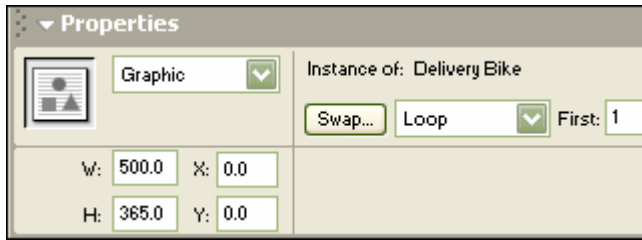


Above: The registration point will be in the center of the object if you do not change it here. The black square shows the location of the registration point relative to the entire area of the symbol.



Above: By clicking the box in the upper left corner of the 3x3 matrix, you set the registration point to be at the upper left corner of the object.

3. Add four seconds' worth of frames to the **Timeline** (your frame rate should be 15 fps, so click Frame 60 in the Timeline, then press F5 once to add 59 frames).
4. Add a keyframe at Frames 15, 30, 45, and 60: Select a frame and press F6; repeat for each of those frames. (If you have a frame rate other than 15 fps, create keyframes at one-second intervals equal to *your* frame rate instead.)
5. Change the *width* of your movie to *half* the width of your photo, and change the *height* of the movie to *half* the height of your photo. For example, if your photo is 400 pixels wide and 300 pixels high, your new movie size should be 200 pixels wide by 150 pixels high. (See Lesson 1, Step 3, if you need instructions for how to change the size of the Flash movie.)
6. Save your file with a new name (such as "phototest_motion.swf").
7. Since you already know how to move an object by dragging it to a new position on the **Stage**, in this exercise you will position the photo by changing its X and Y coordinates. Flash makes this very simple, because when you have *selected* an object on the Stage (click it *once* to select it), you will see the X and Y coordinates for that object on the left side of the **Properties** panel.



Find the X and Y coordinates for a selected object on the Stage by looking in the **Properties** panel.

After changing the X or Y value, press the Tab key to make the new value stick.

Changing an object's position this way is useful because it ensures smooth motion and smooth transitions.

So, first click on Frame 1 in the **Timeline**. Then select the photo on the **Stage** (click it *once* to select it). In the **Properties** panel, double-click inside the field labeled “X” and type “0” (the numeral zero, without the quotes). Press the Tab key.

Now the cursor should be inside the field labeled “Y,” and all the characters in that field should be selected. Type “0” again (the numeral zero, without the quotes), and press the Tab key.

Pressing the Tab key is *necessary* because otherwise the new value does not always “stick.” Sometimes Flash adds seemingly random tenths to the value, but pressing Tab ensures that you get round numbers.

You should be able to see that the *upper left* corner of your photo is now exactly aligned with the upper left corner of the **Stage**.

Note: If you did not set the registration point for this symbol to the *upper left corner* of the symbol (see Step 2 above), this will not work properly. The registration point of the symbol will be at the X and Y coordinates shown in the **Properties** panel.

The illustration below shows what will be visible in the final Flash movie—the photo will slide up, then left, then down, then right.



Frame 1. X: 0, Y: 0. Frame 60 is the same as this one.



Frame 45. X: -249.0, Y: 0. The photo is 500 pixels wide.



Frame 15. X: 0, Y: -181.0. The photo is 365 pixels high.



Frame 30. X: -249.0, Y: -181.0.

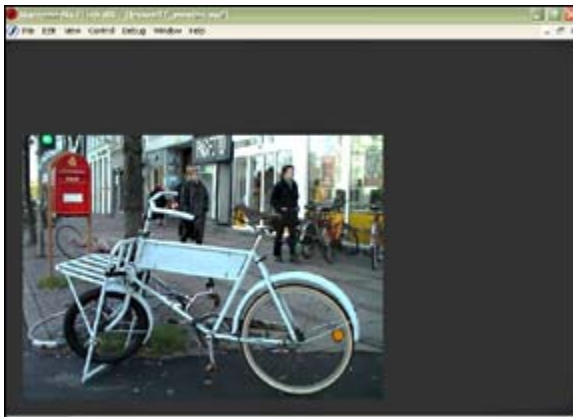
8. Click in Frame 15 in the **Timeline**. Now you will reposition the photo to align the *lower left* corner of the photo with the lower left corner of the Stage. (After you apply the motion tween, the photo will appear to slide straight down.) The “X” coordinate stays the same, because X is the horizontal position of the object, and you are not moving it horizontally yet.

The “Y” coordinate needs to be the *negative* of half the *height* of your photo, plus 1. If your photo is 300 pixels high, for example, change the “Y” value to -149. Y represents the top-edge position of this object (because the *registration point* is at the top edge). You are moving the object upward from 0, and that is why Y is a negative number. When an object’s registration point is below 0, the Y coordinate is a positive number.

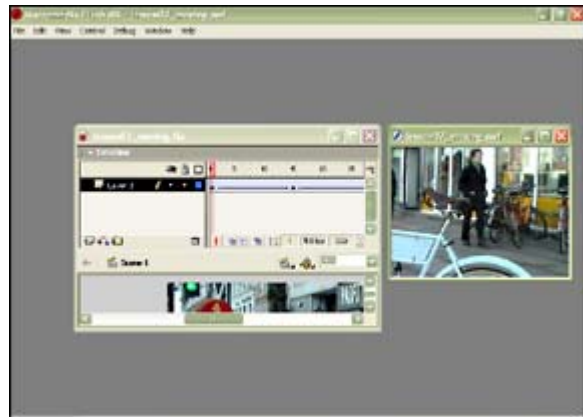
Zoom in (use the Zoom tool) on the bottom edge of the photo and check to make sure you have covered the Stage.

9. Apply a motion tween to the first sequence in this movie. That is, click once in the **Timeline** anywhere between Frames 1 and 15 and select “Motion” from the **Properties** panel. (See Lesson 2, “Animating Two Symbols,” Step 5, if you need a refresher.)
10. Save and test your movie (Ctrl-Enter or Cmd-Return). If your Flash application window is “maximized” (full screen), the test will not look very impressive. You’ll see

the *entire* photo moving in that case—but remember, your **Stage** is actually *smaller* than the photo. If you “restore” the *Flash test window*, you’ll get an accurate impression of how the motion works, because you’ll see the movie at its actual size.



Above: When you test your movie, the default view may be full screen. This is not the real size of your SWF.



Above: In “Test Movie” mode, if you “restore” the screen (instead of “maximize”), you will see the true size of your SWF (the window on the right).

- Time to move on to Frame 30, the third keyframe in your **Timeline**. Here, you will reposition the photo to align the *lower right* corner of the photo with the lower right corner of the Stage. (After you apply the motion tween, the photo will appear to slide leftward.) The “Y” coordinate must match the one in Frame 15, because Y is the top-edge position of the object, and that should not change in this frame. In other words, the vertical position remains the same, and so Y must be the same.

The easiest way to set this up is to click on Frame 15 in the Timeline, click once on the photo, double-click in the “Y” field, and copy the numeral there. Then click Frame 30 in the Timeline, click once on the photo, double-click in the “Y” field, and paste.

The “X” coordinate here in Frame 30 needs to be the *negative* of half the *width* of your photo, plus 1. If your photo is 400 pixels wide, for example, change the “X” value to -199. X represents the left-edge position of the object (because the *registration point* is at the left edge). You are moving the object left from 0 (or backward, in a sense) and that is why X is a negative number. When an object’s registration point is to the right of zero (or forward), the X coordinate is a positive number.

Axis	Direction	Coordinate Represents	Negative	Positive
X	Horizontal	Horizontal position of object’s registration point relative to the Stage	Registration point to left of zero point	Registration point to right of zero point
Y	Vertical	Vertical position of object’s registration point relative to the Stage	Registration point above zero point	Registration point below zero point

Zoom in (use the Zoom tool) on the right edge of the photo and check to make sure you have covered the Stage.

12. Apply a motion tween to the second sequence in this movie. That is, click once in the **Timeline** anywhere between Frames 15 and 30 and select “Motion” from the **Properties** panel. *As an alternative*, you can right-click (Windows) or Control-click (Mac) in the Timeline and select “Create Motion Tween” from the pop-up menu. You may find this a more convenient way to apply a motion tween.
13. Save and test your movie (Ctrl-Enter or Cmd-Return). Remember the size concern described in Step 10.

It’s okay that this movie is looping. There is no need to change that.

14. Frame 45 is the last place where you need to change the coordinates, and in fact, you need to change only the “X” coordinate. You will reposition the photo to align the *top right* corner of the photo with the top right corner of the Stage.

Now that you understand the “Y” coordinate refers to the top edge of this object, you know that Y is 0 if the top of the photo matches the top of the Stage.

So what will the “X” coordinate be? Is the left edge of the photo in the same position as a previous frame? Yes—the photo is sliding straight up from its position in Frame 30. So the value of X for the photo in Frame 45 is the same as the value of X for the photo in Frame 30.

Click on Frame 30 in the Timeline, click once on the photo, double-click in the “X” field, and copy the numeral there. Then click Frame 45 in the Timeline, click once on the photo, double-click in the “X” field, and paste.

15. Apply a motion tween to the third sequence in this movie.
16. Save and test your movie (Ctrl-Enter or Cmd-Return).
17. If you followed the instructions in Step 4 above, the X and Y coordinates of the photo in Frame 60 are both 0. This allows you to complete a continuous loop by simply adding a motion tween in the fourth sequence of this movie. So right-click (Windows) or Control-click (Mac) in the **Timeline** between Frame 45 and Frame 60, and select “Create Motion Tween” from the pop-up menu.
18. Save and test your movie (Ctrl-Enter or Cmd-Return).
19. You should notice a momentary *hesitation* in your loop. This occurs when the loop goes from Frame 60 back to Frame 1—because the two frames are *identical*. While it may seem trivial, this hesitation can appear awkward in some movies, depending on the frame rate and the size of the photo being tweened.

Fixing this is very simple: Click on the frame immediately *before* the final keyframe (in this case, Frame 59) and create a *new keyframe* there (press F6). Then right-click

(Control-click/Mac) once on Frame 60 and select “Remove Frames” from the pop-up menu. This deletes Frame 60, the duplicate of Frame 1.

20. Save and test your movie (Ctrl-Enter or Cmd-Return). Voilà! The hesitation is gone, because now *only one frame* in the loop places the photo at the coordinates 0, 0.

You already know what to do *if the motion seems to be too fast*: Add an equal number of frames to *each* of the four motion sequences (click inside a sequence in the **Timeline** and press F5 several times).

If the motion seems to be *too slow*, remove an equal number of frames from *each* of the four motion sequences (click inside a sequence in the **Timeline** and press Shift-F5 several times).

You can figure out how to apply this technique to many different kinds of photos—especially outdoor scenes. Just *don’t zoom in* on a photo that is already at 100 percent of its actual size; that is the subject of the next exercise.

A note about file size: The example file (illustrated above) was 52.0 KB before any frames or motion tweens were added. Remember that the photo was converted to a symbol immediately, before any frames were added. The final SWF file is 52.2 KB.

Sidebar: Motion Tweens, Bitmaps, and Symbols

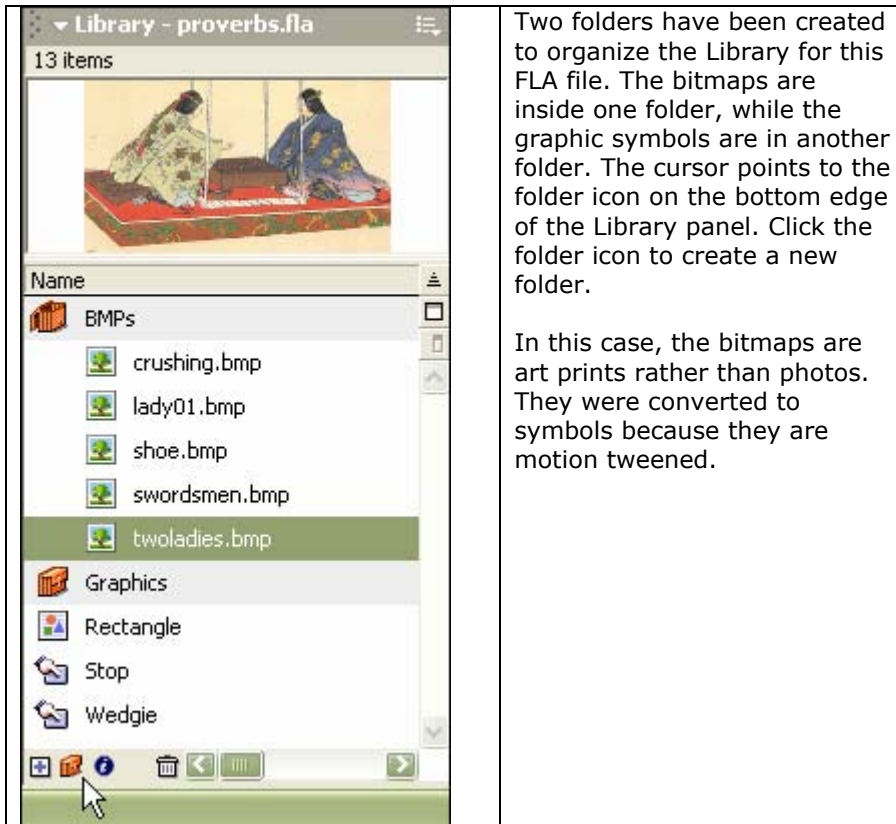
If you place a photo on the **Stage** and then apply a motion tween—*without* converting it to a symbol first—Flash creates a graphic symbol for you. While this may seem convenient, actually it is bad: It will increase the file size (of the final SWF) if you move the same photo more than once.

Each time you apply a motion tween to that “unconverted” photo, an *additional* graphic symbol is created. Flash will name these “Tween 1,” “Tween 2,” “Tween 3,” etc. If you see symbols with those names in your **Library**, you have made this error!

Instead of allowing this to happen, you should convert the photo (or any bitmap) to a symbol (press F8) *immediately* when you first place it on the **Stage**. Afterward, be sure to use the symbol (and *not* the original photo file) *each time you use the image* in your movie. An exception can be made *only if*: (a) the photo never moves, and (b) the photo is used *only once* in the movie. If both those conditions are true, you don’t need a symbol. But in all other cases, you do.

A good practice is to stash your bitmaps in their own folder, separate from all of your symbols, in the **Library**. This helps to prevent mistakes because it hides the bitmaps from you, so you are more likely to use the symbol (as you should). The illustration below shows how to create a folder inside the Library.

Note: Files imported to the **Library** but not used in the movie are not exported in the SWF. With numerous photos in the Library but not on the **Stage** in any part of the movie, the SWF file size is only 1 KB.



Zooming In on a Photo

For this exercise, you need only one photo. Ideally the photo will have a detail worth zooming in to see in close-up.

As you probably figured out when reading about bitmaps at the beginning of this lesson, if you have a photo at 100 percent of its real width and height in Flash and then you zoom in, things will get ugly. Instead of increased detail, you will see bigger pixels.

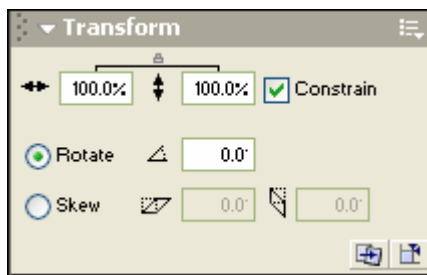
To avoid that result, you must use a photo that is already at the maximum size it will be when zoomed. So, you need to think about the numbers. If you visualize the full-view photo in the Flash movie at 400 pixels wide and 300 pixels high, and you want to zoom in 4x (or 400 percent), you must import a photo that is 1600 pixels by 1200 pixels. That will be a rather large file, and may not be very practical.

Taken from the opposite perspective, you can divide the both width and height of your photo by 4 and use those dimensions as the maximum display size. For example:

Filename	Width x Height (in pixels)	Width / 4	Height / 4
deliverybike.bmp	500 x 365	125 pixels	91.25

So, do your math and decide whether the photo you used before is large enough to use in this exercise. If not, prepare another photo and use it instead.

1. File menu > New (Start with a new, empty file.)
2. Import an *uncompressed* photo file. See Step 2 under “Importing and Optimizing Uncompressed Photo Files” above.
3. Position the photo on the **Stage** (there is only one frame and one layer).
4. Select the photo and Convert to Symbol (press F8): Name the symbol, select “Graphic,” and set the *registration point* to the upper left corner (see Step 2 under “Moving Photos in Flash” above). Zooming will require a motion tween, so you must make it a symbol.
5. Save and name the file.
6. Extend the **Timeline** to two seconds by adding sufficient frames to Layer 1.
7. Make the final frame a *new keyframe* (click that frame and press F6).
8. We’re going to zoom *in*, not *out*, so the *final* frame stays at 100 percent of its size. The photo in the *first* frame must be reduced. Click Frame 1, and then click the photo *once* to select it.
9. Open the **Transform** panel. If you can’t find it, use the Window menu. The *width* and *height* fields in the panel should each read “100.0%” as shown below.



The Transform panel shows the scale of the currently selected object.

10. Double-click into the width field (the one on the left) to select all its contents. Type “25” (without the quotes). Press the Tab key; it will say “25.0%” after you Tab out of the field. The height field should also say “25.0%” after you Tab to it. (That happens because the Constrain box is checked, as shown.) Press the Tab key again to leave the height field. (If 25 percent is smaller or larger than you want for your photo, use a number that works better for you.)
11. Add a motion tween to the frame sequence in the **Timeline**. See Lesson 2, “Animating Two Symbols,” Step 5, if you need a refresher.
12. Save and test your movie (Ctrl-Enter or Cmd-Return). What happens is not pretty (yet), but the basic zoom effect is finished.

13. You need to do two more things to make this look good. The first is to add a mask, on a layer above the photo. The second is to position the photo in the final frame so that the *close-up detail* is where it should be in the zoomed-in photo.

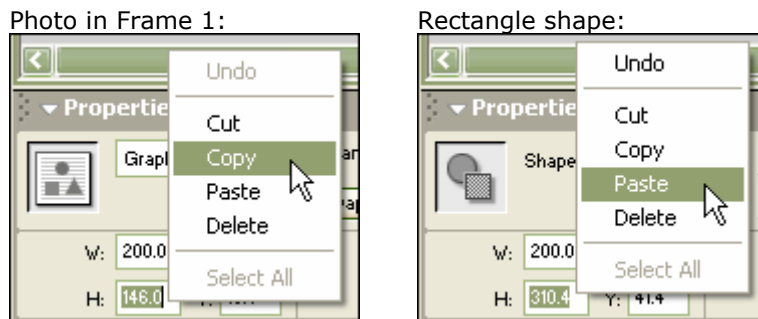
We'll begin with the mask. Add a new layer to the **Timeline** and name it "mask" (it can have any name; it's just convenient to name it "mask"). Name the lower layer "photo," so you remember to maintain good work habits in Flash (always name your layers when you have more than one).

14. Click Frame 1 in the "mask" layer. To create the mask, you must *draw a shape* to exactly cover the small version (Frame 1) of your photo. Initially, this shape *hides* the object you want to be seen, but in the end, the mask shape becomes a window to *reveal* the object.

Select the **Rectangle** tool from the tools panel and draw a rectangle. The color is unimportant, and so is the size (at this time). Delete the "stroke" outline from the rectangle (double-click precisely on the outline to select the entire outline, then press the Delete key)—a mask shape should have only a single fill color.

15. With the rectangle shape selected, change its *width* and *height* in the **Properties** panel to exactly match the width and height of your *reduced-size* photo. After you type the width into the field labeled "W" (on the far left side of the **Properties** panel), press the Tab key to make it stick. Do the same after you type the height into the field labeled "H." (If you don't press Tab, Flash is likely to add some random tenths to your numbers.)

What is the width and height of your *reduced-size* photo? Click it and look at the **Properties** panel to find out. With the photo in Frame 1 selected, you can copy the W value from the **Properties** panel, then select the rectangle shape, and paste the value into the W field in the **Properties** panel. Try it and see—this may seem annoying, but a Flash designer will do this often. It becomes simple if you practice it.



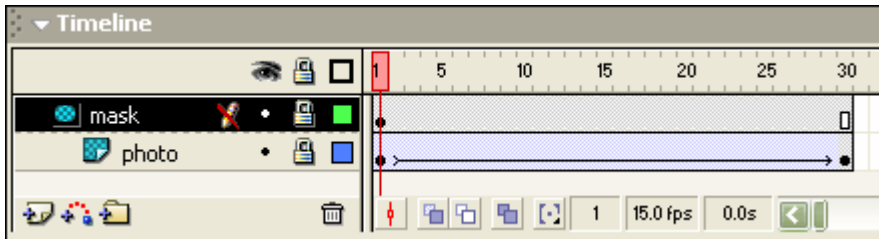
Left: Double-click into the W or H field to select the value. Then right-click or Control-click to get the pop-up menu. (Alternatively, you could press Ctrl-C or Cmd-C to copy the value.) Right: After selecting the object you want to resize, double-click in the W or H field and right-click or Control-click to get the pop-up menu, or use Ctrl-V or Cmd-V, to paste the value. Press the Tab key to make the value "stick."

16. Position the rectangle precisely over your photo in Frame 1. You can use the X and Y coordinates to do this.

Note: If you have “Snap to Objects” enabled (in the View menu), you can grab the rectangle at its upper left corner and drag it to “snap” to the registration point of the photo. This works *only if* both the photo and the rectangle have their *registration point* in the upper left corner.

17. Here’s how you make the mask effect happen: Right-click (or Control-click) on the “mask” layer in the **Timeline**. Select “Mask” from the pop-up menu.

Your **Timeline** should look similar to this now:



Both the “mask” layer and the “photo” layer are *locked*. In this state, any shape within the mask layer is invisible, and it hides everything outside its edges in the masked layer below it.

Notice that the masked layer (labeled “photo”) is indented beneath the mask layer (labeled “mask”). This indicates that the mask will work correctly.

Notice also that both the “mask” layer and the “photo” layer are *locked*. The lock icon to the right of the layer name shows this. While the two layers are locked, the rectangle shape is invisible and the mask is in effect while you are editing the movie. You can test the motion now by simply pressing Enter or Return.

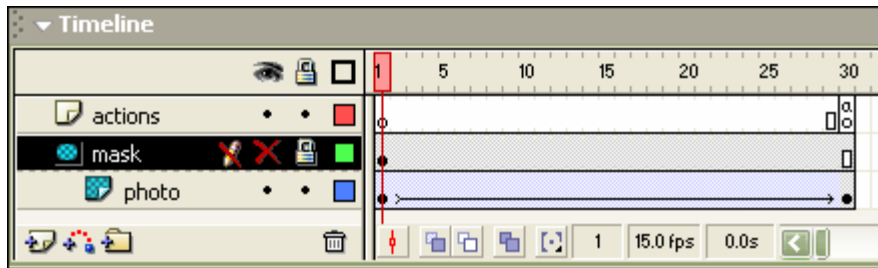
While the layers are *locked*, you cannot make any changes in those layers. You can *unlock* a layer by clicking the *lock icon* to the right of the layer name. To *lock* it again, click the *black dot* that replaced the lock icon.

18. Add a new layer, name it “actions,” and place a `stop()` ; action in the final frame. See Step 20 under “Animating Two Symbols” in Lesson 2 if you need a refresher.
19. To see the effect of the mask, save and test the movie (Ctrl-Enter or Cmd-Return).
20. The last step is to position the photo in the final frame so that the *close-up detail* is where it should be in the zoomed-in photo. If by chance the detail in your photo is in the upper left corner, you’re already finished—because that corner is what you see in the final frame. But if the detail is in *any other part* of the photo, you need to reposition the photo in the final frame.

First, however, you must unlock the “photo” layer so you can change it. To *unlock* the layer, click the *lock icon* to the right of the layer name.

As soon as you *unlock* the “photo” layer, the rectangle shape (on the “mask” layer) becomes visible again. To *hide it* in editing mode, click the *black dot* to the right of the layer name (“mask”) below the *eye icon* in the **Timeline**.

Note: Hiding a layer in editing mode in Flash has *no effect* on the final SWF file.



The “mask” layer is still *locked*, but the rectangle is not visible. Its invisibility is indicated by the red X beneath the eye icon. Without that X, you would see the rectangle shape covering your photo on the Stage. The “photo” layer is *unlocked*, as indicated by the black dot beneath the lock icon.

21. Now you can reposition the photo in the final frame of your movie. To see whether you have it right: (a) click the X in the “mask” layer and (b) click the *black dot* under the *lock icon* in the “photo” layer to restore the mask effect. If you need to reposition the photo again: (a) click the *black dot* under the *eye icon* in the “mask” layer and (b) the *lock icon* in the “photo” layer to make the “photo” layer editable again.



Here is the result of a photo at 40% of its original size (left) zoomed it to 100% (right). Notice that the final frame has been repositioned so that the woman walking is the focus, not the red mailbox that occupies the upper left corner of the full image.

22. When you are satisfied, save and test your movie (Ctrl-Enter or Cmd-Return). If the zoom seems too fast or too slow, add or remove frames to alter the speed.

Note: There is another way to handle the positioning of the zoom detail, using the **Free Transform** tool. In some ways, it is simpler, but when you had finished all the steps shown above, you would then need to reposition *everything* on both layers relative to the Stage, and that is more complicated than the method described here. (It would not be complicated if the entire **Timeline** were inside a *movie clip*, because in that case you could position the movie clip anywhere on the Stage.) If you’re interested in exploring the other method, see the Flash Help files under “Working with Graphic Objects”: Transforming objects > Working with the center point during transformations.

Fading Photos into Each Other

This exercise explains how to accomplish a cross-fade between photos. To follow the instructions, you should have three related photos saved as *uncompressed* files (BMP or

PICT). Crop each photo to the same width and height. Each photo shown below is 320 pixels wide and 240 pixels high. The movie size is 360 x 280, allowing a 20-pixel margin all around the photos. This way, the pictures can be positioned precisely on top of each other, each in a separate layer in the Timeline.

1. File menu > New (Start with a new, empty file.)
2. Set the frame rate to 15 fps. This will make it easier to follow the instructions.
3. Import your three photos to the Library.



Photo 1: 320 x 240,
226 KB



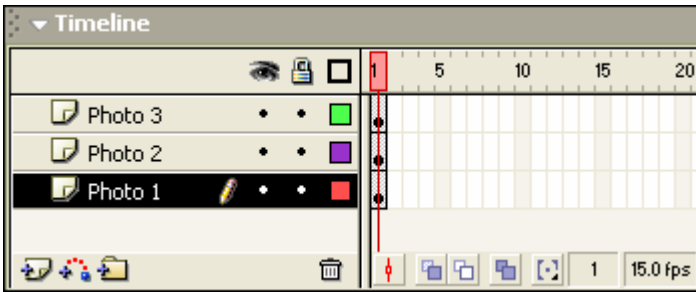
Photo 2: 320 x 240,
226 KB



Photo 3: 320 x 240,
226 KB

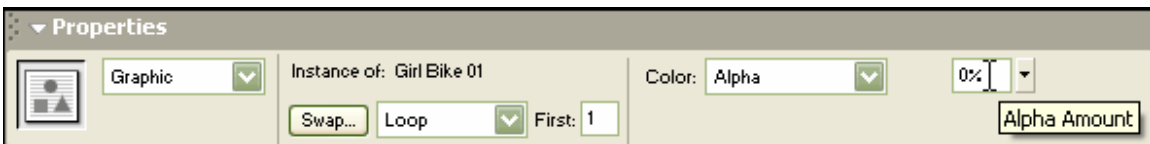
With JPEG Quality at 90% in the Publish Settings dialog, the final SWF file using these three photos is 91 KB.

4. Create two *new layers* in the **Timeline**. Name the bottom layer “Photo 1,” the middle layer “Photo 2,” and the top layer “Photo 3.” You will stack the photos in reverse order because Photo 2 will cover Photo 1, and then Photo 3 will cover Photo 2.
5. Select Frame 1 in the layer “Photo 1,” and drag your first photo onto the Stage.
6. Select the photo and convert it to a graphic symbol (press F8). Make sure the *registration point* for this object is in the *upper left corner* (see Step 2 under “Moving Photos in Flash” above). Give the symbol a descriptive name (this makes it easier to manage the **Library**). One method is to give the symbol the same name as the imported photo file, e.g. “Delivery Bike” is the graphic symbol containing *deliverybike.bmp*.
7. With the photo selected, in the **Properties** panel, change the X and Y coordinates to position this photo exactly as the other photos will be positioned. For example, if I have a 20-pixel margin all the way around my photos, both my X and my Y will be 20.
8. Repeat Steps 4, 5, and 6 for the other two layers, placing the appropriate photo in each layer. Make sure to *convert each photo* to a graphic symbol!



Photos that fade over each other must be in separate layers.

9. Save your file. No need to test it yet, but it's always a good idea to save after you have spent some time positioning objects.
10. Now you need to make the photos fade in and out. Begin by extending the Timeline. To do that, select Frame 45 in all three layers (Shift-click) and press F5 once. We'll handle the fade effect the same way on each layer, and afterward we'll set up the sequence to show Photo 1, then Photo 2, then Photo 3. In other words, the three layers will be identical until later in the exercise.
11. Create new keyframes in *each layer* at Frames 15, 30, and 45.
12. Click in Frame 1 in the "Photo 3" layer, then click once on the photo to *select* it. Look at the far right side the **Properties** panel and find the menu labeled "Color." Open the menu and select "Alpha." Then click into the Alpha Amount field (shown below) and type "0" (zero, without the quotes), then Tab out of the field.



When an object has an alpha of zero, the object is completely transparent. (You can think of *alpha* as transparency.) Each of the photos is going to start out transparent, just like this one; it will be simplest if you work on only one layer at a time.

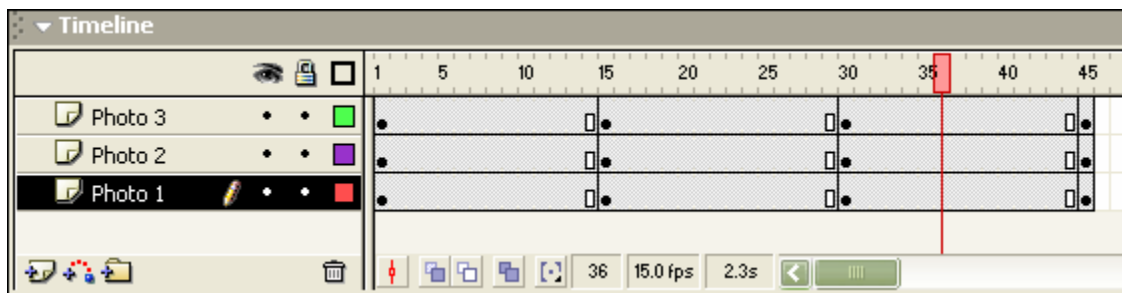
13. Click in Frame 45 in the "Photo 3" layer, then click once on the photo. In the **Properties** panel, open the menu labeled "Color" and select "Alpha." Because the last time you changed alpha, you changed it to "0" (zero, without the quotes), it should automatically be "0" now. (If it's not, just type "0" again.)
14. Click in Frame 15 in the "Photo 3" layer, then click once on the photo. In the **Properties** panel, open the menu labeled "Color" and select "Alpha." This time (in the middle of the layer), you need the photo to be fully visible, or *not transparent*. Normally that would mean "100%" in the Alpha Amount field—but not when you are fading bitmaps in Flash. If you use "100%" you will usually see an ugly effect in the final movie, so standard professional practice is to use "99%" instead. Type "99" (without the quotes) into the field, and Tab out of it.
15. Finally, click in Frame 30 in the "Photo 3" layer, then click once on the photo. In the **Properties** panel, open the menu labeled "Color" and select "Alpha." Because the

last time you changed alpha, you changed it to “99” (without the quotes), it should automatically be “99” now. (If it’s not, just type “99” again.)

16. Save the file. Just to be cautious.
17. Repeat the procedures in Steps 12 through 15 for the other two layers. It will work best if you *lock* the “Photo 3” layer to work on “Photo 2,” and then *lock* both “Photo 3” and “Photo2” to work on “Photo 1.”

To *lock* a layer, click once on the *black dot* under the *lock icon*, to the right of the layer name. The black dot will be replaced by an identical lock icon. To *unlock* a layer, click the *lock icon* to the right of the layer name. When a layer is *locked*, you cannot edit it. It’s very useful to lock a layer when you need to work on a layer below it—it can prevent mistakes!

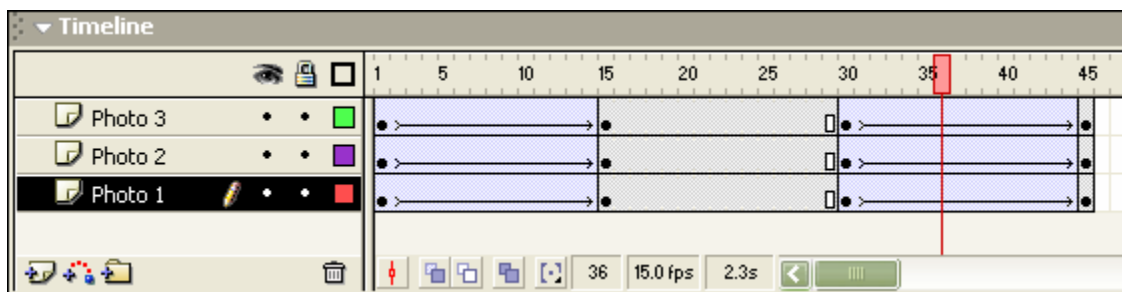
In Frames 1 and 45 in each layer, the *alpha* for the photo should be 0. In Frames 15 and 30 in each layer, the *alpha* for the photo should be 99. Your **Timeline** should look like this now (all layers have been *unlocked*):



The Timeline above is ready for you to apply the motion tweens. Notice that all the layers are unlocked.

18. Next, add a motion tween between Frames 1 and 15 on *each* layer. This will cause the photos to fade in, from 0 alpha to 99 alpha. That may not seem like “motion,” but it definitely requires a motion tween.

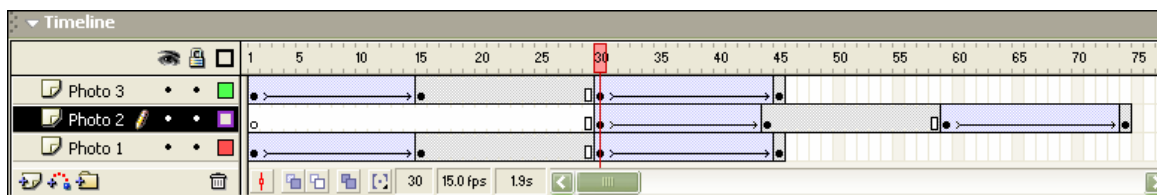
Add a motion tween between Frames 30 and 45 on *each* layer. This will cause the photos to fade out, from 99 alpha to 0 alpha. Your **Timeline** should look like this:



The Timeline above is ready for you to set up a staggered sequence for the three photos.

19. Save the file. If you want to test it, go ahead—but it won't be pretty. All three photos will fade in and then out at the same time.
20. Now you need to *stagger the layers* so the photos fade in one at a time, not all at once. Since the layer “Photo 1” contains the first photo, it does not change. Start with the layer “Photo 2”: *Select* the entire layer by clicking on the *layer name*.
21. With the entire layer *selected* (it will be black, to show that it is selected), *click and hold* on Frame 1. That is, position the mouse cursor over Frame 1, press the button, and don't let go! While holding the button down, *slide toward the right*, being careful to stay inside the same layer. (Don't move up or down in the Timeline—just drag to the right.) When the first frame is at about the Frame 30 position, *release* the mouse button.

What you should see now is a blank, empty, white sequence of about 15 frames at the beginning of the “Photo 2” layer. If the first frame of *the first motion tween* in that layer is *not* at Frame 30, add or remove frames *in the white sequence* to fix that. You are working to maintain 15-frame intervals for everything in this movie so that all the timing is equal.



The Timeline looks like this after you have moved all the frames in the “Photo 2” layer, according to the instructions in Step 21.

22. Now you must do the same thing with the layer “Photo 3”: *Select* the entire layer by clicking on the *layer name*. Follow the instructions in Step 21, but with one difference—*drag this sequence* all the way to Frame 60.
- It should be clear to you from looking at the **Timeline** now that the image in the layer “Photo 1” will begin to fade in immediately. When it begins to fade out (Frame 30), the image in the layer “Photo 2” will begin to fade in. When the image in the layer “Photo 2” begins to fade out (Frame 60), the image in the layer “Photo 3” will begin to fade in.
23. Save and test your movie (Ctrl-Enter or Cmd-Return).

You'll see that you need a `stop()` action in a new layer above the final frame, Frame 105, because the loop does not look very good—after two nice cross-fades, the final photo fades away completely, and only when it is gone, the first photo fades in again.

What if you want to *loop* these fading photos and maintain the same fade effect (smoothly) when the loop returns to the beginning? That's what you will do next.

Looping a Fade Effect

Start with the file from the last exercise, completed through Step 23.

1. Add a *new layer* above the three you have, and name the new layer “actions.” That new, empty layer should automatically fill 105 frames (the length of your movie).
2. In the “Photo 2” layer, click Frame 76 and add a *blank keyframe* (press F7). Then click Frame 105 in the same layer and fill that space with empty frames (press F5 once). This *extends the layer* to match the two above it and is considered good practice by professional Flash designers; it helps you understand your **Timeline** more clearly when all the layers are the same length.
3. The “Photo 1” layer is now the only short layer. We’re going to insert blank frames up to Frame 90, but not after Frame 90. So, add a *blank keyframe* at Frame 46 (click Frame 46 and press F7). Then *extend* the blank frames to Frame 89 (click Frame 89 and press F5).
4. Save the file.
5. In the “Photo 1” layer, double-click in the first motion sequence (between Frames 1 and 15 in that layer). Without moving the mouse, *right-click once* to get a pop-up menu. From the menu, *select* “Copy Frames.” This is a fantastically useful operation to know!
6. Click in Frame 90 in the “Photo 1” layer (really, there is no frame there yet, but that’s all right). *Right-click once* to open the pop-up menu, and then select “Paste Frames.”
7. You get a broken motion tween (you know it’s broken because the line is dotted, not solid), but that’s easy enough to fix. Click once in the last frame, at the end of the dotted line, and then *press F6 once* to add a regular keyframe. The motion tween is no longer broken.
8. What you just did was to copy the *fade-in sequence* for your first photo and duplicate it directly under the *fade-out sequence* for your third photo. This should be starting to make sense now—the first photo will be fading in while the last one is fading out. Great! But what will happen when the movie loops back to Frame 1? The same photo will fade in again! That would look awful—so we are about to prevent it from happening.

Where you really want the playhead to go after Frame 105 is the place *where the first photo is already fully visible*: Frame 15. To make it go there, we will *label* that frame, in the “actions” layer. So click once on Frame 15 in the “actions” layer and make a *new keyframe* (press F6). Then click into the Frame Label field in the **Properties** panel and type “restart” (without the quotes). Press Tab to leave the field and make the frame label stick. If you need a review of frame labels, see Step 3 under “Buttons That Let You Jump on the Timeline” in Lesson 5.

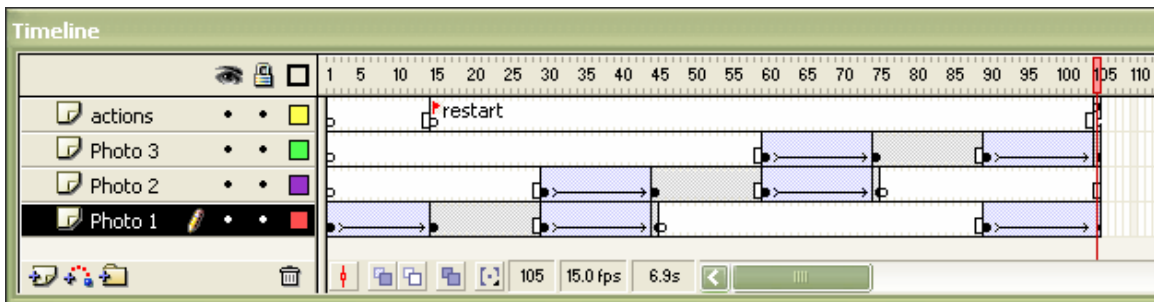
9. All that remains is to tell the movie to go to the frame labeled “restart” when it loops instead of to Frame 1. So, click Frame 105 in the “actions” layer and open the **Actions** panel (press F9). In the actions list on the left side of the panel, open the sections “Actions” and then “Movie Control” by clicking each one once. Under “Movie Control,” find “gotoAndPlay” and *double-click it*. Change “Type” to “Frame Label.”

Open the “Frame” menu and *select* “restart.” Your ActionScript should now look like this:

```
gotoAndPlay("restart");
```

If you need a review of this procedure, see Steps 10 and 11 under “Buttons That Let You Jump on the Timeline” in Lesson 5. In this case, you have written the ActionScript on a *frame* instead of on a *button*. That means the action will be executed as soon as the movie reaches that frame.

10. Save and test your movie (Ctrl-Enter or Cmd-Return). You should have a perfect loop with identical cross-fades every time. Your **Timeline** should look like the one shown below.



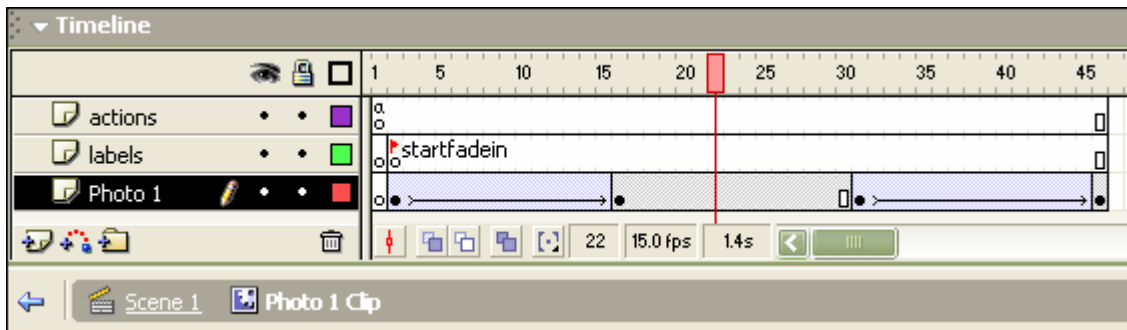
This timeline looks complex, but only because the photo layers have been staggered. Each sequence lasts 1 second (15 frames).

Sidebar: Doing It All with Movie Clips Instead

Now that you understand the staggered timing of the cross-fades used in the previous lesson, think about how much easier it would be to do all this with movie clips instead. Don't panic—it *really would be easier*.

First, imagine each photo as a separate, individual movie clip, like the ones you built in Lesson 6. Each photo would have its own separate **Timeline** in that case.

The Timeline for *each photo's movie clip* would look the same:



The ActionScript in Frame 1 is a simple `stop()` action, nothing more. The frame label in Frame 2 is just like the frame labels you have used in earlier lessons. Notice that Frame 1 in the “Photo 1” layer is an *empty* keyframe.

The big difference—and the newest idea for you, probably—is that the three separate movie clips would *communicate* with each other. At the point in the clip where you want the next photo to fade in (always in Frame 30, if all clips have the same Timeline), you would add a new keyframe (in the “actions” layer, of course) and write one line of ActionScript in that keyframe:

```
_parent.photo2_mc.gotoAndPlay("startfadein");
```

When a movie clip plays and reaches the end of its Timeline, it loops back to Frame 1. In this case, you wrote a `stop()` action on Frame 1, and so it will stop there and wait—until this Timeline is set into action again by *another* movie clip!

There are two concepts you already know in that line of ActionScript, and one new concept:

- “_parent” refers to the Timeline that *contains* this movie clip. Flash must look *outside* this movie clip to find the other movie clips (the ones holding the other photos). This word “_parent” is what tells Flash where to look. This is a new idea introduced here for the first time: When ActionScript refers to an object *outside* a movie clip, it must refer to it in a way that shows the position of the outside object *relative to* the movie clip where the ActionScript is written.
- “photo2_mc” is the *instance name* given to another movie clip that is outside this one. You learned about giving an instance name to a movie clip with the sliding panel in Lesson 6, in Step 17 of the exercise “Movie Clip No. 3: A Pop-up Panel.”
- “startfadein” is the frame label in that *other* movie clip, with the instance name “photo2_mc”—don’t forget that you are going to have all the movie clips constructed the same way, so the label on Frame 2 will be the same word in each movie clip.

This concept is somewhat complex, so to save space in print, no more will be said about it here. On the Web site for this book, in the **Lesson07** folder, find the FLA file named *photo_mc_fades* and download it to see exactly how these movie clips are used. You’ll see that the parent movie’s Timeline is only two frames. The first frame contains a basic preloader script so that the photo movie clips all have time to load.

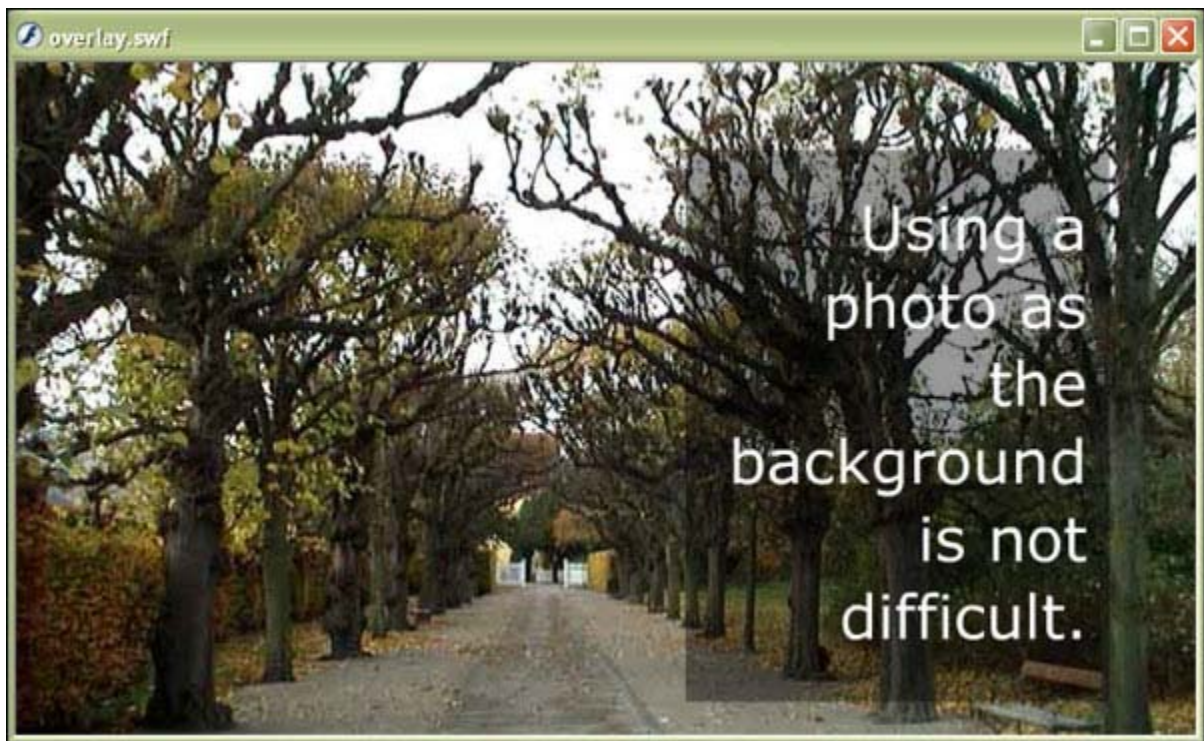
While the example uses only three photos, it would be just as easy to use more (although you would need to consider the increase in file size). This file (88.6 KB with three movie clips) is actually 2 KB smaller than the one with the same three photos (90.7 KB) used in the previous exercise.

Using a Photo as the Background

If you would like to use a photo as a *full background* for an entire Flash movie, the main things to consider are the *size* (width and height) of the Flash movie, the *size* of the photo (both file size and dimensions) and the *position* of the photo.

1. Determine the size of your Flash movie. Say it is 700 pixels wide and 400 pixels high. In that case, you need to have a photo that is 700 x 400, or slightly larger. (Alternatively, you may match the size of the Flash movie to the size of the photo.)

2. Import the photo file to the Library (File menu > Import to Library).
3. If there are already other layers in your movie, create a *new layer* in the **Timeline** and drag it to the bottom of the stack of layers. Name it “background” (without the quotes).
4. Click once on Frame 1 in the new layer and drag the photo from the **Library** to the **Stage**. There is no need to Convert to Symbol if this photo never moves.
5. Position the photo carefully so that it aligns with, or covers, the edges of the **Stage**.
6. If you add more frames to layers above the “background” layer, make sure to also add frames to extend the “background” layer out to the same place in the Timeline.



On the Web site for this book, in the **Lesson07** folder, find the SWF file named *photo_overlay* and play it to see exactly how to use a photo as a background image for multiple frames in a Flash movie. The SWF demonstrates how to use semi-transparent color overlays to tint the background photo or to make text easier to read. (The FLA file is also provided.) If you use a color overlay, you can probably reduce the file size of the SWF significantly by reducing the quality of the bitmap (the photo), and the overall look of your movie will not suffer.

Sidebar: Importing a Sequence of Images

Flash will do a lot of work for you, if you give it a little help. If you have an image sequence (for example, 12 photos that illustrate how to pitch a baseball):

1. Name the files sequentially, e.g., *pitch01.jpg*, *pitch02.jpg* ... *pitch12.jpg*.
2. Click once on the one (empty) frame in the **Timeline** where you want the sequence to begin.
3. File menu > “Import” (*not* “Import to Library” in this case).
4. Select just the first image file, and then click Open.
5. Flash will ask you if you want to import the sequence. Click Yes.

Flash will automatically put each image in a new keyframe, in order, in the same layer. This can save you a lot of time. If you use uncompressed image files instead of JPGs, you can compress them all with one step (see Step 8 under “Importing and Optimizing Uncompressed Photo Files,” above).

If you add a *new layer* with two buttons, and apply the ActionScript `prevFrame()` to one button and `nextFrame()` to the other, you will have an instant slide show.

Working with Photos Summary

Using photographs online always demands that the producer balance the *quality* and the *screen size* of the images with the *technical capabilities* and *attention span* of the intended viewers of the photographs. Even if you are delivering the photos via Flash on a CD or DVD, you must consider the width and height (in pixels, not inches) of the users’ screens. You can always scale a photo down (to a smaller width and height than actual size) in Flash, but if you scale it up, it will not look good anymore (because it is a bitmap, not a vector image).

You can now move, zoom and fade photos smoothly and professionally, using little or no ActionScript. You can adjust the image quality of a photo to a precise degree and make your own decisions about the unavoidable tradeoffs between image quality and file size (which affects the user’s download time). You can avoid having Flash create a lot of unnecessary symbols for you, which would increase the file size of the final SWF.

You have encountered several new aspects of the Flash production environment, such as the X and Y coordinate fields, the width and height fields in the **Properties** panel, the registration point of a symbol, masks, and alpha (transparency). All these features and properties give you the ability to control the movement of photographs in Flash. The “lock/unlock layers” and “show/hide layers” options in the **Timeline** enable you to work competently with layers and build more sophisticated Flash movies.

At a minimum, you should be able to bring photos into Flash and make certain they look as good as you want them to look, and yet not require even one more kilobyte of memory than absolutely necessary.

Conclusion

In this lesson, you have learned to:

1. Differentiate between a bitmap and a vector image.
2. Plan the width and height of your Flash movie relative to the width and height of your photos.
3. Import photos (or other images) to the Library.
4. Align an image relative to the Stage using the Align panel.
5. Open the Bitmap Properties dialog from the Library
6. Change the Quality setting for an individual bitmap image.
7. Swap one symbol for another (Properties panel).
8. Change the JPEG Quality setting for all uncompressed bitmaps in the Publish Settings dialog.
9. In the Bitmap Properties dialog, determine whether to choose "Photo (JPEG)" or "Lossless (PNG/GIF)" for compression type.
10. Convert a photo to a graphic symbol.
11. Change the registration point for a symbol.
12. Move a symbol by changing its X and Y coordinates to ensure straight, steady movement.
13. Correct for hesitation in a looping animation by removing one frame.
14. Avoid having extra symbols (Tween 1, Tween 2, etc.) created when you move a photo.
15. Create folders in the Library to organize your assets.
16. Plan for zooming in on a photo.
17. Using the Transform panel to zoom in or out.
18. Create a mask effect to hide part of an image.
19. Lock and unlock individual layers in the Timeline.
20. Show and hide individual layers in the Timeline.
21. Work with the alpha (transparency) property of a symbol.
22. Use 99% alpha instead of 100% when fading photos.
23. Move all the frames on one layer.
24. Stagger layers to allow a smooth fade effect between photos.
25. Copy and paste entire frames, or sequences of frames.
26. Create a smooth, continuous loop of photos that fade into each other.
27. Use a large photo as a background image in a Flash movie.
28. Place a complete sequence of images to the Timeline in one step.