CHAPTER 1

Core Terms and Concepts

Any Illustrator user who plans to make graphics for the Web needs to be aware of some basic ideas and options. For example, you must take into account how you control the space for a graphic in an HTML document, what file formats are available to you, and how download time affects the site visitor's experience. Within Illustrator itself, you can choose among various ways of putting graphics into a web page. Which workflow you use will affect the choices you make as you create graphics. We'll refer to these issues throughout the book. If you already have a clear sense of them, consider skipping ahead to Chapter 2, "Essential Illustrator Tools and Techniques."

This chapter covers the following topics:

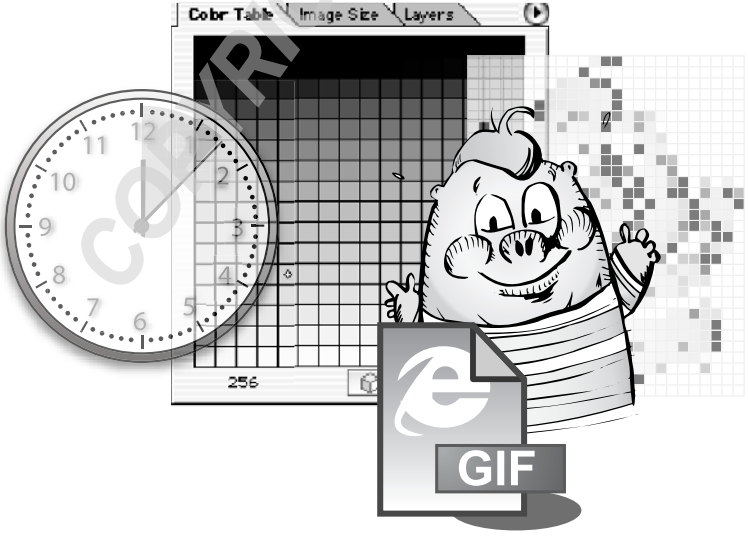
Graphic space in HTML

File formats

Transfer times

Color issues

Using Illustrator in a web workflow



Web Terms and Concepts

This section introduces the core issues involved in working with graphics in HTML pages: how graphics can be placed on a page, the various web file formats, download times, and the "web-safe" color limitations.

Graphic Space in HTML

Users of Illustrator are accustomed to being able to place objects exactly where they want them. If you want an object in a specific position on a page, you just put it there. This is not how things work in HTML. HTML was designed to transfer text data; it was never intended to do the work it is doing now, and that includes handling robust page design. Objects flow in HTML as in a text document. Things start in the upper left and flow to the right and down. Graphics appear inline, as though they were characters in a word. This means that to place objects in specific locations, you'll need to work around HTML with one of four basic workflows (listed here in order of preference).

Tables

What they are: *A basic HTML workaround for controlling graphic space.*
What you should know: *The majority of web pages use tables because they are widely understood by web browsers. They are a bit clunky compared with CSS layers (see the next section, "Cascading Style Sheets (CSS)").*

The most common way to control graphic space is by using a table. These tables are similar to the ones you'd create in a word processor or spreadsheet. You divide a rectangular area into boxes. Each box, or *cell*, can contain text, graphics, or even other tables. You can make the table large enough to cover an entire page. The contents of each cell are expanded, contracted, or combined with other cells to push and pull items into the position you desire. By nesting tables together and adjusting the amount of area cells take up, you can place items where you want them to be. Figure 1.1 shows the difference between simply placing an image on a page and placing it within a table.

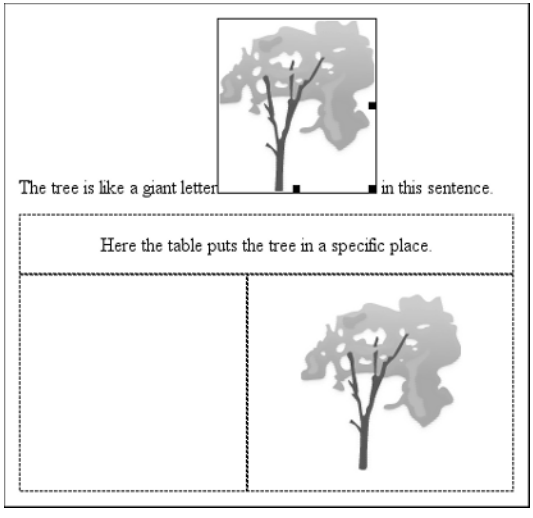


Figure 1.1
By default, objects in an HTML page flow like text. Tables enable you to set items in a specific position.

Tables are useful and common, but they have limitations. Objects can't be on top of other objects, and they may not appear the same in different browsers. These issues aren't showstoppers, but you need to be aware of them. Illustrator can generate tables for you automatically. For more information, see Chapter 8, "Creating Complete Pages."

Cascading Style Sheets (CSS)

What they are: *These style sheets form a modern structure for describing graphic space using x- and y-coordinates.*
What you should know: *CSS is understood only by 4.0 or later web browsers; they feature richer design opportunities than tables.*

Cascading Style Sheets (CSS) create a model for defining graphic space that is understood by 4.0 and later browsers. This means site visitors must have Netscape 4 or Internet Explorer 4 or later to view pages that use CSS. Other contemporary browsers, such as Safari, Camino, and Phoenix, also recognize CSS. Current statistics place the number of users with pre-4.0 browsers at less than 1 percent. This number is still a concern for conservative designers and those responsible for sites that reach broad audiences.

This is not an insurmountable problem. Often, sites designers create two versions of websites: one for those with modern browsers, and one for those without. When users load the site's home page, they are sent to the version of the site that's appropriate for their browser.

Why would you go to these lengths? Because style sheets enable you to do things you can't with standard HTML. Namely, you can position objects where you want them, make objects overlap each other, show and hide the items with scripting, and animate objects. Most of the drop-down menus that you see in web pages (like the one shown in Figure 1.2) are created using CSS. As time passes, this model will likely surpass tables as the de facto choice for web page creation. Like any technology, CSS can be misused to create visual noise and incompatible web pages, but it is very promising.

Figure 1.2
Drop-down menus illustrate the way CSS objects can overlap other contents.

CSSs are commonly referred to as *layers*, or *floating boxes*. They can be used in conjunction with tables or in lieu of them. Illustrator can generate them for you; you'll learn more about working with Cascading Style Sheets in Chapter 8.

Frames

What they are: *Frames use different HTML pages to break up graphic space; this is a mostly passé model.*
What you should know: *Frames are often used to separate navigation items from page content. They are used less and less frequently as other technologies obviate their need.*

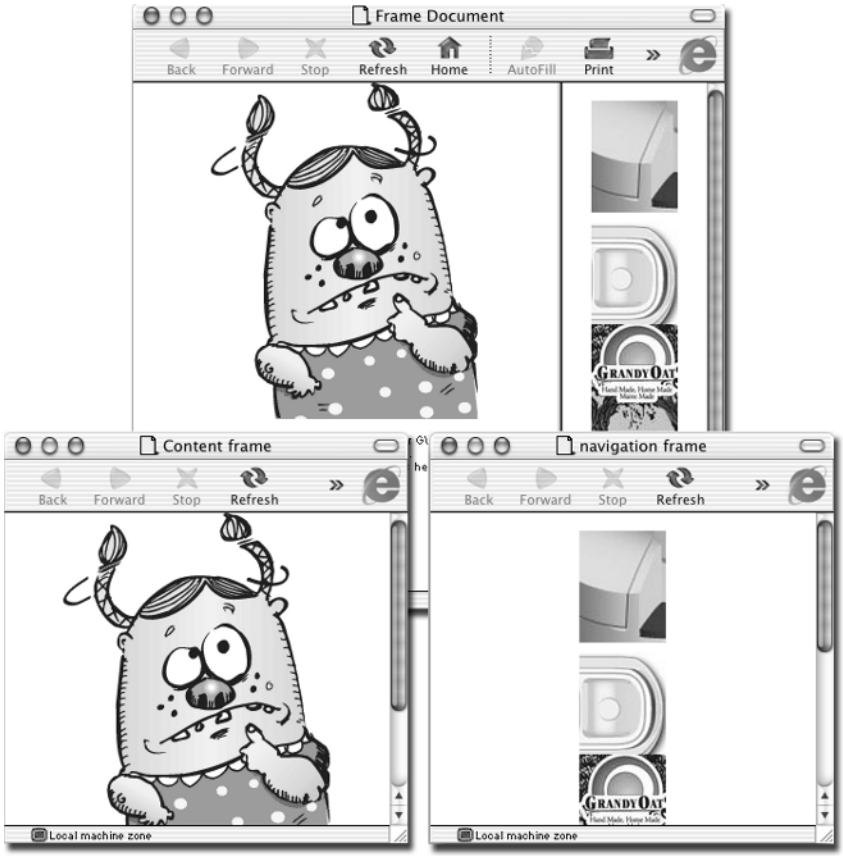
Frames are HTML documents that contain other HTML documents. This breaks space up into rectangular space, as shown in Figure 1.3. Often, users set up common

items in a single document, creating links that load in another frame. This doesn't exactly solve the positioning issue, but it can be used to block out documents. Because frame documents can contain other frame documents, you could conceivably nest enough of them together to position objects exactly. And, of course, framed documents can use layers or CSS.

Frames have other problems, though. Their structure prevents them from being indexed properly, making it harder for users to find your page. In addition, they can't be understood by screen readers. When the visually impaired experience a web page, it is read to them by screen-reader software. Frame documents contain only references to other HTML pages, so there is no text to read. This flummoxes the software, rendering the page unusable to a portion of your potential audience.

Illustrator doesn't make frames. Likely you shouldn't either.

Figure 1. 3
A frame document and its component pages



PDF Files

What they are: *Files that will appear exactly as you designed them in Illustrator. They can be used instead of HTML or as part of a page.*
What you should know: *PDFs require a plug-in in order for them to be viewed in a browser. This can cause hiccups in the user experience.*

PDF files can appear in a browser exactly as you designed them in Illustrator (Figure 1.4). Given this, they could be used to set up entire pages, complete with the font of your choice. That is seldom the case, however. Instead, PDFs are used almost exclusively for downloadable versions of print documents.

Some people feel that requiring a plug-in has prevented PDF files from being more widely used as web content. The truth is more likely that the viewing experience is just different enough to irritate users. You have to wait for the plug-in to load, the Acrobat interface loads by default, and you need a special tool to select text. Until Acrobat 5, PDF files also couldn't be understood by screen-reading software. So while it's possible to use PDFs to design graphic space, it's rarely done. The reasons in this case concern individual preference more than technical limits.

File Formats

When you save art, you need to assign a format to it. The format encodes the picture in a specific way. This may alter the way the file appears and may change what you can and can't do to the file later.

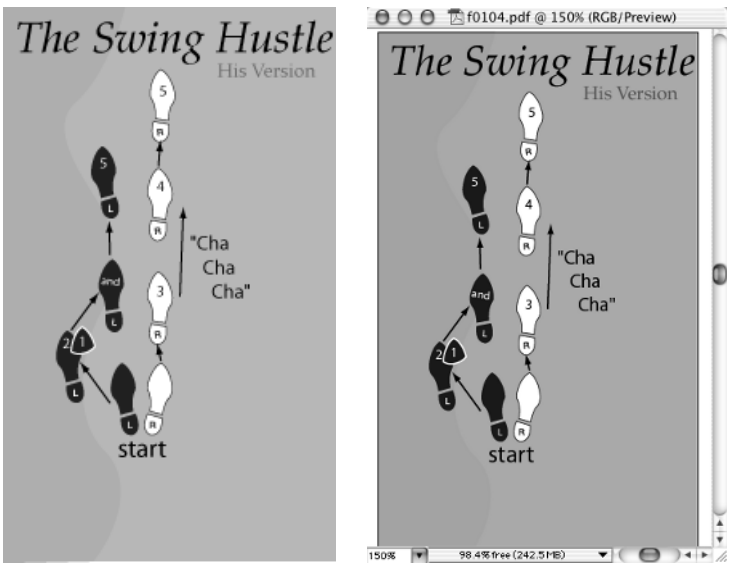


Figure 1.4
A PDF file (left) in Illustrator and then (right) used in a web page.

You should consider two major aspects: the way graphic space is described and how color is handled. Let's look at the most basic difference first—pixels, which are often also called "bitmap" or "raster" art, and vectors.

Pixel-Based (Bitmapped) vs. Vector Graphics

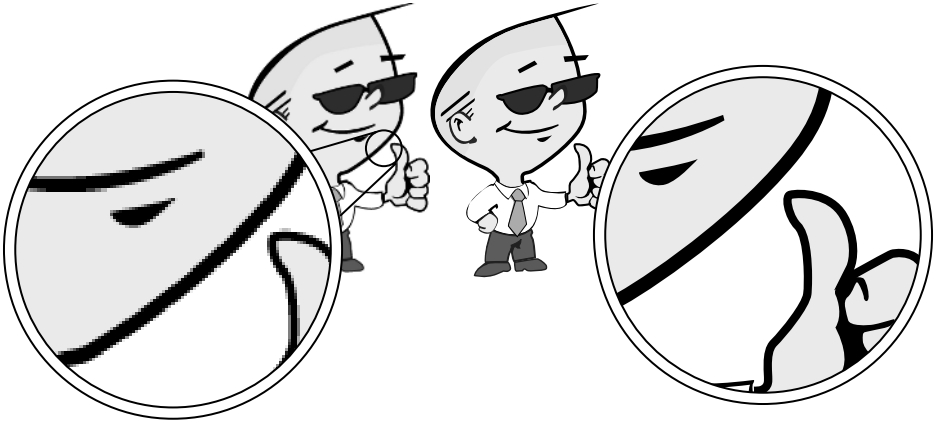
Web graphics may be either pixel-based (bitmapped) or vector based. Illustrator is one of the few applications that can create both. Pixel graphics are the most popular online choice. Pixels are tiny, colored squares. The file is composed of a grid of these squares. Together they create the appearance of the image. The larger an image is, the more pixels are required to describe it and the larger the file size is. Pixel images are optimized to a specific physical size and, once optimized, work best at that size. HTML tags can change the size at which the image is displayed, but this usually results in poorer image quality or inefficiency (Figure 1.5). Additionally, the requirements for displaying pixel images are different than those for printing them. This means that viewers will be able to see your images clearly, but will not get great results if they attempt to print the images.

Pixels can be saved into several file formats, and each has its own set of requirements (which we discuss in the sections that follow). Several of these formats are extremely popular online because web browsers understand them right out of the box. The HTML tags used with pixel images enable you to add links and image maps to graphics, increasing their functionality.

Vector is the model Illustrator uses to describe graphics by default. When you are working in Illustrator, you are most commonly creating vector art right on the page. As you may know, vectors create art with a connect-the-dots model. Points in space are identified and lines are drawn between them to create shapes (Figure 1.6).

Vector graphics are great because they can be sized up and down without affecting edge definition or detail. Edges in these graphics are sharp, and type can even be retained so that it may be copied or searched. Normally, when you make raster art, text becomes a picture of the characters. It's locked in that format and you can't highlight the word to

Figure 1.5
Pixel graphics scaled up (left) suffer detail loss. Vector graphics (right) do not.



select its content. With Scalable Vector Graphics (SVG), you can. You can select text, or even assign it to a different font on the fly. These graphics may even be printed at high resolution. Vector graphics may contain links, animation, and additional functionality, but these features must be added in the application that authors the graphics, rather than in the HTML that surrounds them.

Several vector file formats can be understood online by web browsers. The Shock Wave Flash (SWF) and SVG are the two most commonly used. Unfortunately, browsers don't have the ability to display these graphics out of the box and require a plug-in to view them. The plug-ins are free, and modern browsers come with the Flash plug-in already installed. A recent study by Macromedia (creators of the SWF format) indicated that over 92 percent of current web surfers have the Flash plug-in installed. Still, many designers are concerned that requiring users to download a missing plug-in will drive them away from their site. In addition, the SVG plug-in doesn't have anywhere near the penetration that Flash does.

When deciding whether to use pixels or vectors, take the plug-in into account. Consider your audience: the more technically sophisticated they are, the more likely they are to already have the plug-ins required. You may be developing for CD or intranet distribution, where you can control the environment the page is viewed in. You should also consider whether you intend to author entire pages in vectors or integrate them with HTML items and pixel graphics. Often, designers will isolate the use of vector graphics to self-contained vector pages that open as separate windows or offer complete alternate websites for non-vector-enabled websites. You may simply decide that the advantages of vector graphics outweigh the risk of inconvenience and proceed.

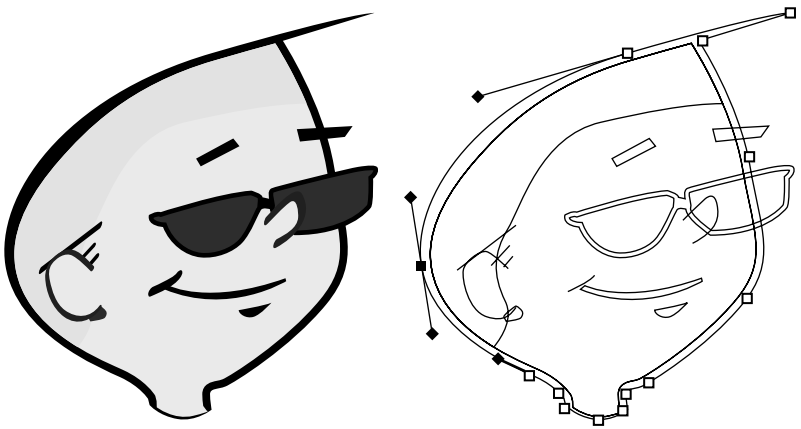


Figure 1.6
Vector graphics describe space with points and lines instead of pixels.

Color

Different file formats support different color models. This means that the act of saving to a particular format switches the color in the image to the model the format understands. This is mostly an issue when saving pixel graphics. Each pixel must be set to a specific color. The act of changing each pixel may alter the appearance of the image overall.

Colors are measured in terms of the amount of memory assigned to each pixel. The more memory that is allotted, the larger the number of colors that can be used. Memory is measured in bits. Bits are computer units that are typically grouped in sets of eight (the byte). An 8-bit image can support up to 256 color choices. Grayscale art uses this model of color. Twenty-four-bit color uses three *bit channels*, or color components. The famous *RGB* (Red, Green Blue) color space uses this model.

How pixels are assigned each color is also an issue worth discussing. In an 8-bit format, each pixel is assigned to one of the 256 colors available in the document. In practice, you'll have some power over which specific colors are used. In full RGB color, you won't be choosing specific color sets, but you will still have to make choices. Files are smaller, and transfer over the Internet faster, if pixels are similar to each other. To make files smaller, pixel values are changed in the course of saving them. You will have to balance between speed and quality of your image. Further, as you save over a web graphic again and again, the degradation keeps adding up. This is not a function of bit depth, but it's good to be aware of that up front.

Formats

In this section, we'll examine each of the popular file formats.

GIF

What it is: *An 8-bit pixel based file that supports animation and transparency.*
What you should know: *Graphics Interchange Format (GIF) is likely the most common file format you will be saving from Illustrator. The GIF was originally developed by CompuServ (an early AOL-style online service).*

GIFs are typically used for logos, buttons, and art with areas of flat color. It's a common format for Illustrator to create because the strengths of the format often dovetail with the kinds of images created in Illustrator. GIFs are also commonly used as *spacer graphics*—placeholder objects that help control graphic space inside a table (see the earlier section "Tables"). Illustrator may produce these automatically as it generates HTML, but it is rarely done manually.

GIFs use an 8-bit color model. This means that each pixel must be assigned one of 256 possible colors. You need not use all 256 colors. In fact, using fewer colors reduces the

overall file size. The set of possible colors that each pixel can call from is called a *color table*. You can define color tables for different graphics to help optimize image quality. Still, reducing the colors in a file down to this number may seriously hurt the appearance of the art, as you can see in the bottom two images in Figure 1.7. Understanding the process of defining and applying color tables is a major part of creating GIFs.

Because GIFs use a limited number of colors, they tend to be best suited for graphics that have large areas of flat color rather than ones with a lot of color changes, like photographs. Illustrator files often consist of solid areas of color, making GIFs a common choice for their optimization. The presence of gradients, gradient meshes, feathers, drop shadows, or raster art in an image may result in a file better served as a JPEG, though, as you can see in Figure 1.8. You should certainly consider all options as you optimize files.

GIFs support both sequential animation and transparency. Many of the animated banner ads that you see on the Web today are GIFs. You can't produce animated GIFs directly out of Illustrator, but you can set up objects to be animated in another application. In fact, Illustrator is often an excellent choice for setting up animation because of such features as blends and brushes. For more information, see Chapter 9, "Creating Animations."

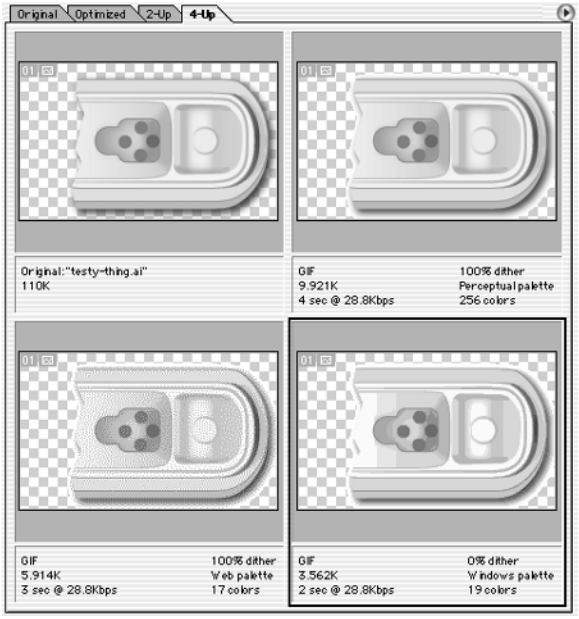


Figure 1.7

The same graphic optimized with different color tables appears very different.

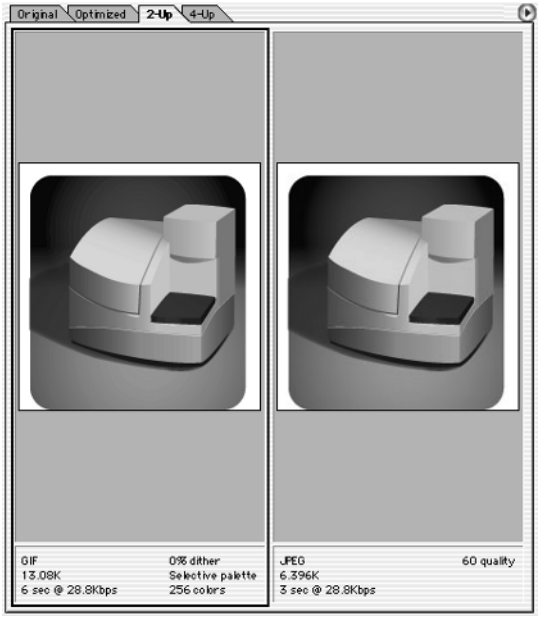


Figure 1.8

Although the image has areas of solid color, it is too large when optimized as a GIF (left) because of the gradients. Optimized as a JPEG (right), it reduces in size.

Illustrator supports one hundred levels of opacity. GIFs support one. Pixels in the GIF format must be either opaque or transparent. You won't be able to make pixels semitransparent and build them atop other images in a web page; Figure 1.9 shows what happens when you try. Nonetheless, this is a fine feature and enables you to create objects with negative space. This is often used to display background images behind art (Figure 1.10) or to give the appearance of a non-square image.

GIFs may be *interlaced*, which means they load in stages. As shown in Figure 1.11, they start as blocky images and resolve themselves over time. Users are given a sense of the item at first, but it may take a few passes to see what it really is. This is intended to make the loading of pages quicker. It does reduce the load time overall but can be frustrating when used on important or large images. In addition, this approach increases the file's size slightly. For this reason, interlacing is usually reserved for nonessential graphics.

Figure 1.9

Partially transparent objects in Illustrator (left) are flattened in the optimized GIF (right). This results in images that appear correct relative to one another but do not continue to be transparent against other graphics on a page.

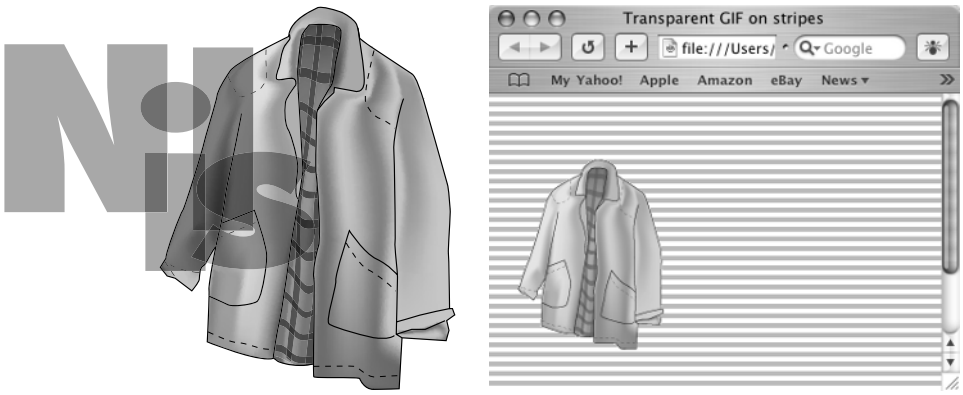
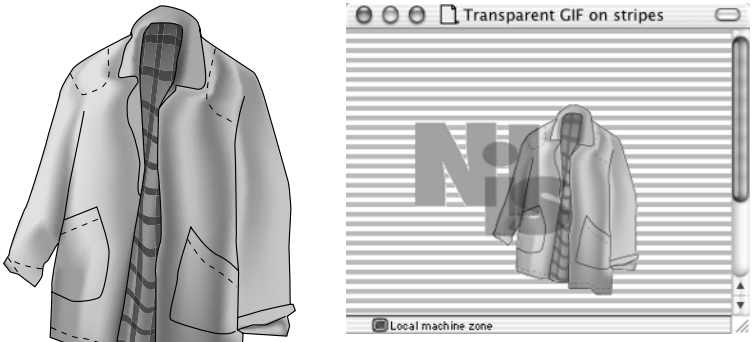


Figure 1.10

A transparent GIF by itself (left) and used in a web page with a background image (right)



JPEG

What it is: *The file format most commonly used online for photos and art with continuous tone.*
What you should know: *The Joint Photographic Experts Group (JPEG) format is used for both print and web.*

Assuming that all JPEGs are web-ready is a mistake. The format supports multiple color models (including both RGB and CMYK) and high resolution. CMYK images can't be viewed in web browsers, and images are displayed at screen resolution, making high-resolution images gigantic. You can avoid both of these issues if you are making JPEGs from Illustrator by choosing File → Save For Web. Do not confuse this command with the File → Export method for making JPEGs, which you'd use to create print graphics.

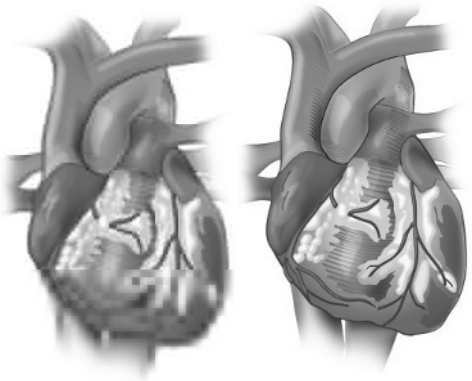


Figure 1.11
An interlaced graphic loads in several passes, appearing blocky at first (left) and then becoming clearer until finishing (right).

LOSSY AND LOSSLESS COMPRESSION

All compression algorithms are defined as either *lossy* (like JPEG) or *lossless (like GIF)*. As the name implies, lossy compression methods actually lose some information; that is, they discard pixel data that the algorithm determines to be redundant, often by changing pixel values if they differ only slightly from neighboring pixels. Color values are lost, not resolution. This displays itself as ugly artifacting in some JPEGs. The effect is cumulative. That is, each time you open and save a JPEG the compression is applied again, further degrading the image. In this sense JPEG is often used as a verb as in "that file has been JPEGed too much".

Lossless compression methods achieve their compression without discarding any color data. The most common web example of lossless compression is GIF. GIFs can only hold up to 256 colors, so in that sense, color from your image may be "lost" as you save for web, but the effect is not cumulative each time you re-save the optimized graphic. In that sense the compression is "lossless".

What you should take away from this is that you should always save your Illustrator files along with the web graphics. Having the source graphic enables you to create a fresh version of the art. If you must edit a web graphic, be aware that repeatedly saving JPEGs will eventually wreck them. For full details on compression, see Chapter 3.

JPEGs use 24-bit color (RGB). Images are simply RGB; no color tables are needed. JPEGs are well suited for photographs and images with continuous tones. When you are creating graphics in Illustrator that have a lot of gradients, meshes, or soft-edge effects (such as Gaussian Blur), you should consider JPEGs.

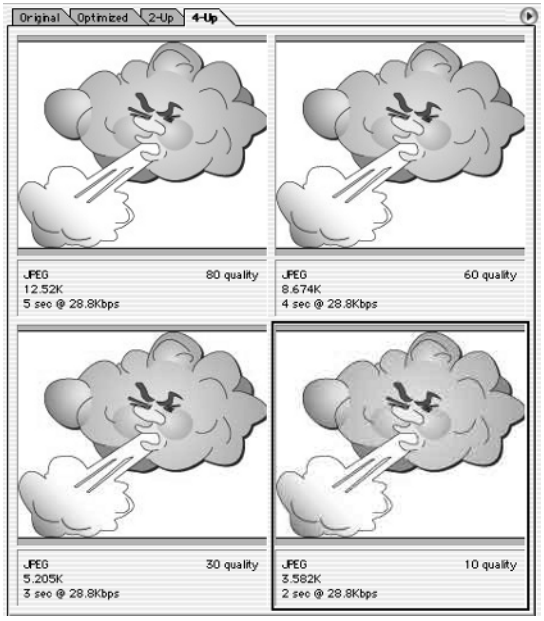


Figure 1.12
JPEGs often produce ugly artifacts near edges. Here we see the same image with progressively stronger compression. As the compression increases, the defects become more pronounced.

JPEGs use a *lossy* compression model. When making JPEGs, there is a direct connection between the amount of compression and the amount of damage done to an image (Figure 1.12). In photos, you may not notice this damage, especially at higher settings. In solid images, though, it often results in noticeable artifacts about the edges. You can mitigate this effect somewhat, but you may end up with unrecoverable files.

JPEGs do not support transparency or animation. The only special feature they support is interlacing. Interlaced JPEGs are called *progressive* JPEGS. As with interlaced GIFs, progressive JPEGs load in a series of passes, each one more greatly resolving image detail. Very old browsers may not support progressive JPEGs. Unsupported graphics display as an empty box with an "X." For more information, see Chapter 7, "Optimizing Spot Illustrations."

PNG

What it is: *An alternate to both GIF and JPEG.*
What you should know: *The Portable Network Graphic (PNG) is a promising format that is not fully supported by all browsers. For this reason, most users avoid the format unless they are working in a completely controlled viewing environment, such as a CD or intranet delivery. It's unfortunate that the format is not widely accepted, because it has distinct advantages.*

PNG has been the "format to watch" for several years now. Delightfully, it supports full transparency, rather than the one level of transparency GIFs use. Combined with CSS layers, PNGs could make web design more like traditional page design. The problem is the way that transparency is handled. Internet Explorer versions 5 through 6 for Windows (the most dominant browser on the market) renders only one level of transparency. This makes your nice soft drop shadows look fine on the Mac and in Netscape but lousy on the PC. Although this is no different than GIFs, it is highly frustrating and can lead to complications. Many users avoid the PNG format principally out of old habits or latent distrustfulness. You can check in on the PNG format to see the current level of browser support at `www.libpng.org/pub/png/pngstatus.html`.

Illustrator can write two PNG formats: PNG-8 and PNG-24. The "8" and "24" refer to the bit depth of the graphics created. For simplicity, you can think of the 8 and 24 versions of the formats as alternatives to GIF and JPEG, respectively.

The PNG-8 format uses a color table just as GIFs do. PNG-24 supports full (RGB) color. For more information, see Chapter 7.

Both PNG formats support transparency. The PNG-8 version features the same transparency as GIF files. PNG-24, however, features a full 256 levels of transparency, as illustrated in Figure 1.13. This means web graphics can have the same transparency as the source Illustrator objects, even against other objects in a page. Combined with CSS floating boxes, this is a potent combination.

Both PNG formats may be interlaced. PNGs do not support animation.

SWF

What it is: *An online vector file format commonly used for animation, logos, and interactivity.*
What you should know: *The ShockWave Flash (SWF) format was developed by Macromedia. It is commonly generated by Macromedia's Flash and has reasonably high browser penetration. Illustrator can make these files for you, but it can't add the interactivity or animation. For this reason, SWF files are usually either spot logos or text illustrations or are continued in Flash or LiveMotion (Adobe's SWF animation application).*

SWF files support full RGB color and can contain both raster and vector art. In fact, some art may be converted to pixels in the saving process. This is one of the things to watch for when you're creating these files, because some Illustrator techniques will create art that converts to pixels automatically. Including pixels will dramatically increase the file's size. For information on anticipating and mitigating this, see Chapter 9. Because SWF files are vector based, they are scalable. This means that you can use the same file at different sizes on the same page without the edges becoming fuzzy (see Figure 1.14).

As we mentioned earlier, SWF files can contain interactivity as well as animation. Because of this, you can create entire websites as SWF files. This approach completely bypasses the problems associated with HTML graphic space. When SWF files were first introduced, a spate of websites appeared that were created in Flash, and this technique was briefly fashionable. Things were flying all over web pages, often for no particular reason. Today, sites still use SWF files, but designers are integrating it in ways that make the site seem more natural and less intrusive to the experience.

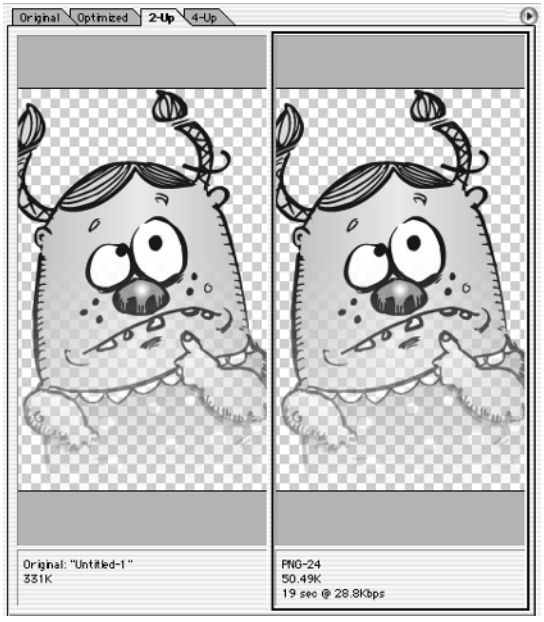
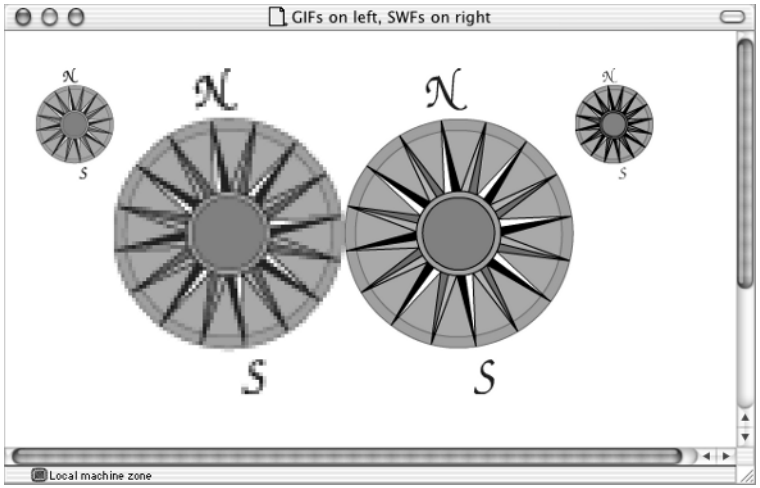


Figure 1.13
PNG-24 supports full transparency. Unfortunately, it is not fully accepted by all browsers.

Figure 1.14
The same (SWF) logo used at different sizes in a web page shows none of the degradation its pixel counterpart does.



SWF files are handled differently than standard graphics in HTML. Macromedia has ownership of the file format and produces separate software (a plug-in) that the browser uses to interpret the files. This strategy enables Macromedia to update its software independently of the browser, but it does create some differences in how the art is handled. HTML tags have attributes. *Attributes* specify such factors as the thickness of a table's border and the color of text. They also enable you to use graphics as links. The HTML tag that is used for standard graphics (`img`) isn't the one used for SWF files (`object`). What this really means is that if you want SWF files to act as links, you'll need to build that into the file and can't add it later in the HTML.

Illustrator is well suited to preparing SWF files and has tools designed for that purpose. But there are specific do's and don'ts related to creating these sorts of files. For a more in-depth discussion, see Chapters 7 and 9.

SVG

What it is: *An online vector file format intended to compete with SWF.*
What you should know: *Scalable Vector Graphics (SVG) is an open-source format that could either be a next big thing or a footnote. Learn more in Chapter 10, "Creating Scalable Vector Graphics (SVG)."*

In many ways, SVG can be compared to SWF, the other online option for vector graphics. Both offer interaction, sophisticated scripting, and animation, and both require plug-ins. SVG has some color- and text-handling advantages over SWF, though. Notably, text in an SVG file can remain live, meaning it can be searched, and copied and pasted as type (see Figure 1.15). SWF files convert text to outline vector shapes. This is a major difference.

Retaining text enables developers to format text on the fly inside an existing file and produces a file that can be read by screen-reading software, providing greater accessibility and compliance to federal laws. Also, SVG is an open format and you don't have to pay royalties to Macromedia (the inventors of SWF).

SVG is based on Extensible Stylesheet Language (XML), and browsers will soon be written to accommodate it without requiring plug-ins. Currently, though, it is not widely adopted and has some limitations on some systems with some browsers. For a full discussion of these issues, see Chapter 10.

Two file formats are associated with Scalable Vector Graphics: SVG and Compressed SVG (SVGZ). The Compressed version can be up to 80 percent smaller than the regular version, but it cannot be edited by a text editor. Both formats use the same dialog boxes when saving.

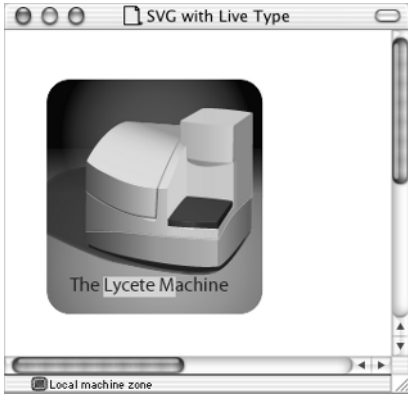


Figure 1.15
An SVG graphic used in a layout may contain live type.

Transfer Times

In most cases, web graphics are transmitted over the World Wide Web. They also see use in intranet models and CD workflows, though, so it's wise to make a distinction. When your art is going to be transmitted over the Internet, you need to factor in the amount of time it will take to download. Slowly loading pages turn users away and is a mark of poor craft. Further, some service providers charge for web hosting based on the amount of data transferred. Seemingly small file size differences are multiplied over the amount of times the file is served. In popular sites, this can translate to a substantial amount of money.

This book devotes an entire chapter (7) to the specifics of optimizing graphics. But for now, be aware that for standard (pixel) graphics, the bigger the art is and the more color it has, the larger the file size will be.

Different web page audiences fall into different demographic groups. This means they may have access to different web technology and will see images and text at different rates. Time, then, is less useful a measure than file size. Although there is room to wiggle on this, try to keep within the following guidelines. Ads and graphics should be no larger than 15k. Buttons and icons should be no larger than 5k. The entire page (including all the graphics) shouldn't be larger than 50 or 60k.

Many designers chafe under file size restrictions. They argue that as newer computers come out, the issue of file size is not as pressing. They feel justified in designing graphics-rich pages with all of the bells and whistles. Remember that the computer the audience has

is only half the issue. Web pages are only as fast as the servers that are transmitting them. I've spent the last year traveling across the country in major metropolitan areas. I have a cable modem at home but use dial-up on the road. Although I have a 56K modem in my laptop, I frequently am unable to connect any faster than 28.8K. This is the case in many cities I visit.

To give you a sense of what these numbers translate to, consider the pipeline that graphics are traveling down. In the next sections, we discuss the technology models your audience will use to see your pages

Modem

Modems use telephone lines to transmit and receive digital data. Speed can be measured several ways, but is usually done in kilobits per second (kbps). Note this is *bits*, not bytes. A byte is eight times as large as a bit. File sizes are measured in kilobytes (k), transfer speed in kilobits (kbps). Most new computers come with 56K modems. This means they are capable of receiving 56,000 bits per second. A slower modem could be transmitting, though, reducing the transfer time. At 56.6kbps, 5k of data transmits in about 2 seconds. At 28.8kbps, it takes about 3 seconds. This means that 15k of data takes around 4 seconds to transmit at 56.6kbps and about 6 at 28.8kbps. Also, 50k takes about 10 seconds to load at 56kbps and about 20 seconds at 28.8kbps. Increasing a page's size to 70k from 50 adds between 4 and 6 seconds to the download time. Sneak up to 80k and it takes a full 30 seconds to load a page at 28.8kbps.

DSL

Digital Subscriber Line (DSL) uses your standard phone line with a special modem. The service is not available in every location. DSL uses part of the phone line, but unlike modems, it doesn't have to tie the phone up so that you can't use it. DSL is typically 3 to 5 times faster than standard phone lines and is one of the two options usually referred to as *broadband*.

There are several types of DSL. Consumers usually get ADSL. The "A" stands for asynchronous, which means the transfer times are different for uploading and downloading. Most ADSL connections let you download at up to 1.5 megabits per second (mbps). (A megabit is a thousand kilobits.) This is great for large downloads. Uploading is slower, typically 128 to 256kbps.The rates will vary, though, depending on your service. You may also be able to get Symmetric DSL (SDSL), which sends data back and forth at the same pace. This is usually 256K to 768K.

A 15k file could download at 1.5mbps in one second. In fact, you could download an 80K file in a second at that pace. At 128 or 256kbps, the 15K file would take 2 seconds. A 5k file transmits in less than a second using any of those options.

Cable

Cable modems transfer data through the cable that brings you cable television. As with DSL, it's much faster than standard cable and the upload times are slower than the download times. Cable can potentially run faster than DSL, but the speed depends on the number of users in your area accessing the service at the same time. Users don't notice this most of the time for day-to-day surfing. The top speed for cable connection is usually 2mbps (although some areas offer 3mbps). Downloads typically clock in around 128 to 384kbps.

As with DSL, our 15k and 5kB files transfer in less than a second. At the top speed of 2mbps, you can download a 100k file in a single second. That would take 40 seconds with a 28.8kbps modem.

Color Issues

Several color terms and concepts are thrown around frequently, and they bear a little discussion here. As with most web issues, you'll need to weigh their importance to your situation and decide what, if anything, to do about them. We'll help you with those decisions in Chapter 7.

Color Naming

In HTML, color is identified either by name or by using hexadecimal notation. You'll see this in the code used to describe text, borders, and the background colors of tables, cells, and layers. Hexadecimal (or *hex*) is the more common model of the two. A finite number of named colors exist, and this model is not used by Illustrator or Photoshop. Illustrator does use hex color values, and it's common to copy and paste the color code between applications.

Basically, hexadecimal is just a way of writing down RGB values with fewer digits. Hex uses base 16 instead of base 10 numbers. This means instead of going from 0 to 9 (base 10), numbers go from 0 to F. In this system, A represents 10, B represents 11, and so on, up to F (15).

To make larger values, numbers are set right to left. In base 10, the second number to the left represents the number of tens. In base 16, it equals the number of 16's. So the number 11 in hex equals 17 (one 16 and one 1). The number F0 equals 240 (fifteen 16's and no 1s).

RGB colors are described in the amount of red, green, and blue light a color is made of. In base 10, the lighting values are measured on a scale of 0 (no light) to 255 (pure bright light). Web colors are rendered in hex with two digits for each color component. The first two are the red component;, then the green and the two on the right are the amount of blue light. For example, the color CC0000 is CC red, 00 green, and 00 blue. This translates to 194 red and no green or blue.

Cascading Style Sheets often identify colors by name. In HTML 4, using style sheets this way is preferable to using color attributes in tags. There are 16 preset colors: Aqua, Black, Blue, Fuchsia, Gray, Green, Lime, Maroon, Navy, Olive, Purple, Red, Silver, Teal, White, and Yellow.

Web-Safe Color

The concept of web-safe color is a controversial one. Here is the basic idea: Different monitors, because of differences in their mechanics, can display different sets of colors. All Macintosh monitors can display, at a bare minimum, 256 specific colors. The same is true for PC monitors, but the sets of colors each can display are not the same. Of those 256, the two sets have 216 colors in common.

These colors are the ones that can be displayed with certainty on every monitor in use today. Collectively, they make up the *web-safe color set*. If a monitor can't display a specific color, it may try to approximate it by *dithering*. Dithering is when pixels of different colors are put next to each other to trick the eye into seeing some other color. This generally isn't a desirable situation—it often looks bad. The safest way to avoid dithering is to use colors you know can be displayed on every monitor. Using these colors doesn't guarantee color veracity. Web-safe colors don't display the same on every monitor—they just don't dither. Further, JPEGs don't use web-safe color. This is only an issue for solid colors, such as HTML text and background colors and GIF files.

Recent studies show that most (over 90 percent) web users have monitors that can display more than 256 colors. Unfortunately, many Windows systems set the default color display to 8 bit, effectively creating low-end monitors from high-end ones. Also, AOL has a feature that compresses art on the fly, limiting the number of colors in the image to 256. Most AOL users have this turned on by default and are unaware of the option. AOL users currently represent about 5 to 6 percent of the web audience.

Hence, the real question is, are there enough AOL and Windows users out there to justify restricting yourself to the (ugly) web-safe color palette? While there is a growing number of artists who feel the answer is no, the issue remains controversial. Better monitors alone have not solved the web-safe problem.

Web colors have hex colors that are stepped in units of 33. Each RGB color must be the 00, 33, 66, 99, CC, or FF hex value in order for the color to be web safe. You can use this to spot a web-safe color at a glance. For example, the color 003CFF is not web safe because the G component (the second set of digits), 3C, is not divisible by 33.

Illustrator in a Web Workflow

This short section is intended to help you place Illustrator into context in a web workflow. What role will Illustrator play as web content is devised, designed, and implemented? Several

tools are available, and Illustrator can wear different hats in terms of its use in graphics creation. You may or may not yet have a clear sense of what you want from the application. Here we'll look at typical web workflows and the role Illustrator may play inside them. We'll also examine the role of other common applications, such as Photoshop and Dreamweaver, and when not to use Illustrator.

Design and Implementation

The first thing to be aware of is that there are two overall functions to creating a website:

- The design of the graphics
- The actual creation of the pages

You can think of these functions as *design*, the shaping of the appearance of a graphic or page, and *implementation*, the construction of the code that delivers it to people's homes. These are two different disciplines with different concerns. Although you may be responsible for both of these things, in many environments the people who design web pages are not the people who actually prepare them. In many large companies, the design department prepares graphics and web pages that the IT department then picks up and adjusts for form and correctness.

The reason this happens is because the IT department has to support the site and is typically responsible for fixing errors. In a complex site with e-commerce and databases, this can be a complicated task. We're taking the time to mention this, because some workflows may not be supported in some environments. Just because a tool exists does not mean its use is permitted or endorsed. The options for Illustrator's use may be limited by your environment. Remember this as you consider your options.

Basic Goals

First, you have to decide what you want to do. Illustrator can generate individual web graphics, graphics that will be converted to web graphics by another application, and complete web pages. Which you choose depends on the particulars of your work environment and what other tools are at your disposal.

Generating Complete Pages

What it is: *Creating HTML as well as graphics when using the File → Save For Web command.*

What you should know: *You can generate complete pages quickly, but they can be hard to edit later and lack some common features.*

Illustrator can generate complete, single HTML pages. When this happens, graphic space is divided using either tables or CSS. Different graphics are generated to represent

each part of the page. Parts of the page can be designated as non-image areas. This enables you to fill in the areas later with HTML objects, such as tables and text. You can even convert existing text to tagged HTML type. Basic interactivity in the form of links and image maps can be created as well. For information on doing this, see Chapter 8, "Creating Complete Pages."

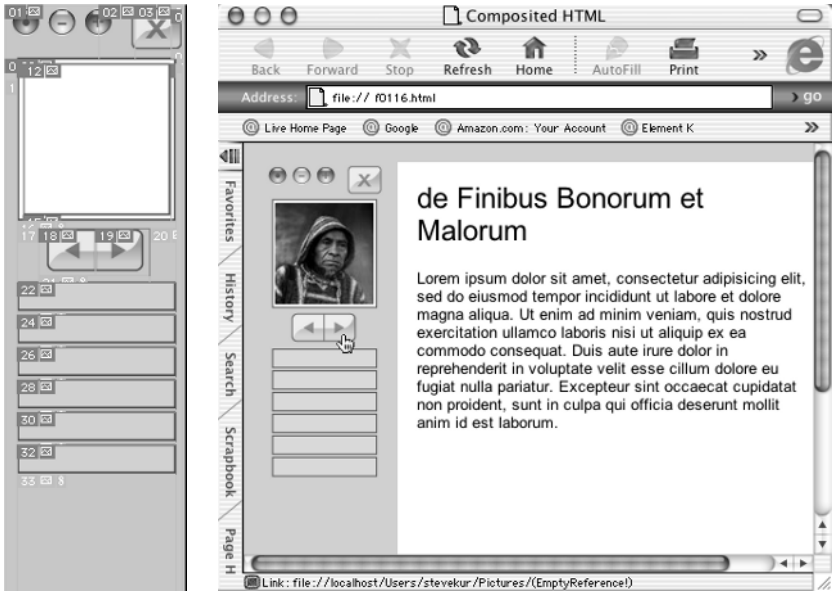
Illustrator is typically not the end of the web page production cycle, however. It doesn't have a rich set of production tools and usually requires additional code massaging to make its pages ready for posting. This isn't a flaw. Illustrator is a graphics creation tool, not an HTML editor. Adobe expects that you will adjust the HTML either manually or in a visual editor, such as GoLive or Dreamweaver.

Editing the pages Illustrator creates can be challenging as well. Notably, you may find the tables it constructs difficult to modify. As users add and subtract cells, the table structure may become confused. It usually takes so much tweaking that it makes more sense to go back to Illustrator and regenerate the page from there.

If you are a GoLive user, generating complete pages also locks you out of a SmartObject workflow (see the next section). For these reasons, Illustrator users typically create complete pages only to comp out a design or to create smaller HTML parts that are then assembled elsewhere. For example, a user may create a navigation interface in Illustrator, save the document as an HTML page, and then place that code in the cell of a table in another document (see Figure 1.16). For instructions, see Chapter 8.

Figure 1.16

Illustrator art is saved as HTML to create a complex table (left). The table is then added to an existing layout (right).



Converting Graphics

What it is: *Creating files that will be converted to web graphics by another application, typically Photoshop, ImageReady, Flash, or GoLive. These are usually either native Illustrator files or Photoshop documents.*
What you should know: *Graphics are typically optimized for the web in other applications to fit into specific workflows, because of user familiarity with other applications or to take advantage of features in other applications.*

Although Illustrator supports a wide array of file formats and web optimization tools, some users prefer to create web art elsewhere. The most compelling reason to do this is that you want to do something you can't do in Illustrator.

A common example is creating animated GIFs and SmartObjects. *SmartObjects* are a workflow device Adobe GoLive uses. The idea is to connect a high-resolution graphic to a web version. This happens when the user places an Illustrator file into a GoLive document. Also, when Illustrator writes HTML it can produce SmartObjects automatically. GoLive walks you through converting the file to a web graphic and then manages the connection. As you make changes to the high-resolution version, the corresponding web graphic is updated automatically. Likewise, you can re-optimize the web version from the high-res version.

GoLive does this by including an extra snippet of code in the HTML. The code is small and can be automatically stripped out when the file is uploaded. Nonetheless, this feature makes some IT departments nervous. They are concerned that GoLive will further rewrite the code they so lovingly created. You should discuss the issue with them and alert them to GoLive's HTML writing preferences. You will also want to be careful about sizing and bounding box issues when using this sort of file.

Animated GIFs are often generated in Photoshop's sister application, ImageReady (free with Photoshop). Users set up an animated GIF by creating a layered Illustrator file and then exporting it to the Photoshop native format. For instructions on this, see Chapter 9.

A third scenario in which you'd want to save graphics for optimization elsewhere is when you are batch-optimizing graphics in an automated workflow. Here, you're optimizing a series of files automatically using a script or an action.

Creating Single Graphics

What it is: *Manually optimizing files for use online, typically through File→ Save For Web but possibly by File → Export.*
What you should know: *You should be aware of a couple of issues involving bounding boxes and edges.*

This workflow is the most common. Here, you're creating vector art in Illustrator and then converting them to web graphics. This is a broad topic, and much of Chapters 3 through 7 is devoted to this workflow. Read on.