

CHAPTER 7

Page Structuring Using Cascading Style Sheets

In Chapter 5, “Web Page Structuring Using Tables,” you learned that using HTML tables to structure the content of elements on the page is not only easy, but also, when used correctly, extremely flexible. Unfortunately, you probably also noticed that as the site became more complex, the process of getting the tables perfectly adjusted and getting the content exactly where you wanted it to be was hit or miss at best. After working with merging and splitting cells, adding transparent GIF spacers, and experimenting with the horizontal and vertical spacing, you might be thinking there’s got to be a better way to structure your site. There is! That is where CSS and, more specifically, AP Elements come in.

As you’ll learn throughout the chapter, AP Elements, or Absolutely Positioned Elements, are nothing more than CSS-driven “content blocks” in Dreamweaver that can contain text, images, media elements, and more. Beyond the simplicities of being able to add content to an AP Element, you can also freely position (usually by dragging and dropping) the AP Element anywhere you want on the page. As you’ll see, this is possible because AP Elements rely on standards-compliant CSS positioning properties for controlling the structuring of these unique elements. Regardless of whether you’re a print designer looking for web-based structuring techniques that closely resemble print programs like InDesign or Illustrator, or a seasoned web page developer who’s traditionally relied on tables for web page structuring, AP Elements are your answer!

IN THIS CHAPTER

- ▶ Introduction to AP Elements
- ▶ Working with AP Elements
- ▶ Designing Tableless Web Pages Using AP Elements
- ▶ Built-In CSS Page Layouts

Introduction to AP Elements

AP Elements in the world of web design mean freedom from messy table workarounds and total control of content layout. To achieve this, AP Elements outline properties for precise placement of elements on the page. Even better, AP Elements offer a third dimension called the *z-index*, ripped from geometric practices based on x, y, and z coordinates. The higher an AP Element's z-index value, the closer it appears to the front of the screen. The lower the z-index value, the further away it seems, resulting in an item closer to the background.

Precise placement of elements on the page? Why would you ever fumble with tabled structures again? This is the question that plagues designers and developers. The answer lies in your user's target browser, which, for the most part, is impossible to know. For AP Elements to be visible in a browser, the browser must be a 4.0 or later version of Netscape or Internet Explorer. For advanced properties outlined by AP Elements to be viewed correctly, the browser must be a newer browser version such as Netscape 7 or later, Internet Explorer 6 or later, Firefox, Safari, and so on. Figure 7.1 shows the Vecta Corp website design using AP Elements in Internet Explorer 7.

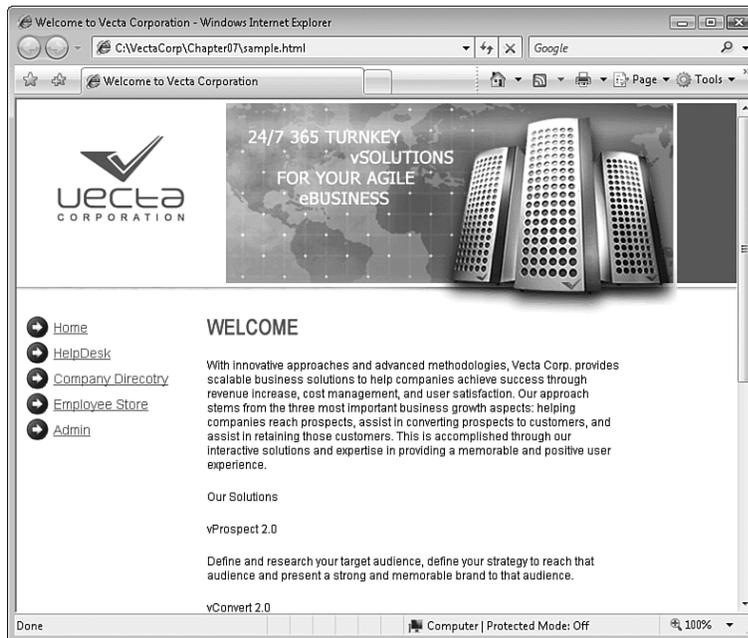


FIGURE 7.1 The Vecta Corp site, designed using AP Elements, shown in Internet Explorer 7.

Internet Explorer 7 is considered a modern browser and supports the newest CSS specifications as they relate to the advanced positioning properties of AP Elements. The outcome would be similar if we viewed the site in Firefox, Safari, Opera, or newer versions of Netscape. However, Figure 7.2 shows the same design in Netscape 4.



FIGURE 7.2 The Vecta Corp site, designed using AP Elements, shown in the Netscape 4 browser.

In this second scenario, the outcome is not as desirable as the first. You can see that the navigation items are falling under the icons, the bitmaps aren't being drawn correctly, and in general, the page looks bad. The reason for this is simple—and lies in the support of AP Elements by older browsers, the introduction of positionable tags with the inception of HTML 4.0, and the code generated by Dreamweaver.

Before we get ahead of ourselves, however, let's backtrack a bit by introducing and discussing the evolution of AP Elements. In the late 1990s, the W3C established a new HTML 4.0 specification that introduced radical changes to the way developers could write HTML, and even better, format that HTML using a new and more robust CSS specification. Combined with JavaScript, the phenomena was coined *Dynamic HTML (DHTML)* and thus introduced designers and developers to a new way of working with web pages. The idea was simple: DHTML was the harmonious combination of HTML, CSS, and JavaScript. Among other things, DHTML would allow developers to change the style declarations of an HTML element by means of JavaScript. Even better, through the use of absolutely positioned elements, until recently referred to as layers, content on the page could be precisely positioned in the browser window using absolute or relative positioning properties. As part of this transition, browsers scrambled to support what was touted as the next "big thing" in web development.

As a result, the two major browsers of the time (Netscape and Internet Explorer) ended up supporting very different extensions to the original specification. For instance, the HTML 4.0 specification brought about two tags used to generate and work with AP Elements: the

`<div>` and `` tags. Although Internet Explorer supported these tags, Netscape 4.0 didn't (in terms of creating AP Elements), instead opting for the proprietary `<layer>` and `<iLayer>` tags. To make a long story short, the dust settled, and a major shift in the browser industry saw users preferring Internet Explorer over Netscape version browsers. The web development trend also moved to support the more popular and much more flexible `<div>` and `` tags. Newer versions of Netscape, Firefox, Opera, Safari, and Internet Explorer now fully support the use of `<div>` and `` tags, and the `<layer>` and `<iLayer>` tags have become deprecated. In fact, Dreamweaver doesn't even support the use of `<layer>` and `<iLayer>` tags, opting instead to create AP Elements using the widely adopted `<div>` or `` tag.

NOTE

Although Dreamweaver no longer supports the `<layer>` and `<iLayer>` tags in its visual environment, that doesn't mean that you can't use these tags. If you're adamant about supporting older versions of the Netscape browser (maybe it's the browser of choice for your organization), you can still find the tags in the code hints menu in Code view. This means that if you plan to support older versions of Netscape and you must use AP Elements in your site, you'll end up coding everything by hand.

In short, you must be aware of your target audience. If your organization is structured so that your livelihood depends on your next online sale, tables are probably your best bet because they are supported by a much greater array of browsers and are backward compatible. If, however, you've developed a new media site to market development services or your aim is to comply with forward-thinking practices, you might opt for a more standards-compliant alternative in AP Elements to impress, show off design and development capabilities, and just plain have a faster loading site. Be aware though, that if the end user is viewing your site with a web browser that is not compatible with AP Elements written using the `<div>` or `` tags, the result will be detrimental to the look of the site, as you saw in Figure 7.2. In the end, it's crucial to be aware of your target audience when using AP Elements in your site.

Working with AP Elements

Now that you have a formal understanding of the power AP Elements are meant to provide, let's actually insert and work with them in Dreamweaver. In the following sections, you'll learn about the various techniques for inserting AP Elements, modifying AP Element properties through the Property inspector and the AP Elements panel, building a tableless web page using AP Elements and CSS positioning properties, and how to use built-in CSS page layouts.

To begin working with the examples in this chapter, you'll need to download the support files online. As you have done for the rest of the chapters in this book, you can work with the examples in this chapter by downloading the files from www.dreamweaverunleashed.com. You'll want to save the files for Chapter 7 in an easy-to-find location. I'll place mine in `C:\VectaCorp\Chapter7`. Also, don't forget to update your site reference in Dreamweaver to point to the newly created folder.

Inserting an AP Element

You have a couple options for inserting AP Elements into a web page; the method you use depends on your needs and skill set. By far the easiest method for inserting an AP Element is to use the AP Element option in the Insert, Layout Objects submenu. To use this method, follow these steps:

1. Create a new HTML page by choosing File, New. When the New Document dialog appears, select the HTML option from the Basic Page category, choose the <none> option from the Layout list, and click Create.
2. Place your cursor on the page and select Insert, Layout Objects, AP Div. The new 200 pixel by 115 pixel AP Element appears on the page similar to Figure 7.3.

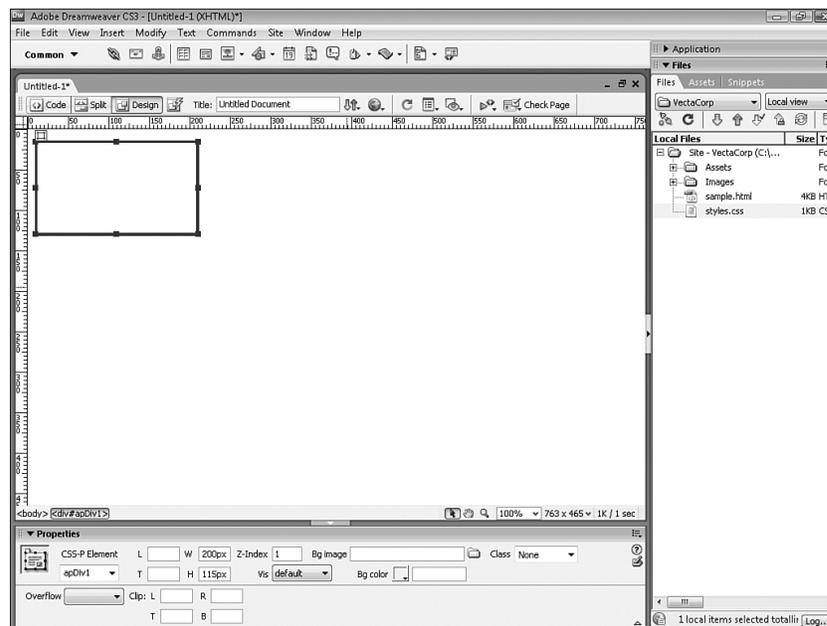


FIGURE 7.3 A new AP Element is added to the page using the Insert menu.

Initially, you'll notice that the AP Element appears as a box with a border surrounding it. You can easily select the AP Element by rolling your cursor over the border of the AP Element until your cursor turns into the Move icon. Now click to select the AP Element. Notice that the AP Element highlights blue and displays small blue squares, otherwise known as *resize handles*, at each point and in the middle of each line. Also notice that a handle appears in the top-left corner of the AP Element, also shown in Figure 7.3.

You can use the small resize handles to click, hold, and drag out to resize the AP Element vertically, horizontally, or both. Furthermore, you can use the top-left handle to select and move the AP Element anywhere you want on the page.

TIP

Dreamweaver also allows you to select an AP Element by its border and move it around on the page. If you use this method, make sure that you don't accidentally select a resize handle, or you'll find yourself resizing the AP Element instead of moving it.

Drawing an AP Element

Another simple alternative for adding an AP Element to the page is to use the Draw AP Element method. With this method, you can easily click and draw an AP Element in the page to any dimensions that you want. To use this method, follow these steps:

1. Switch to the Layout category of the Insert bar.
2. Select the Draw AP Div icon, the second icon to the right of the Standard and Expanded buttons in the Insert bar.
3. Place your cursor on the page, click, hold, and drag out to draw the AP Element in the Document window.

The AP Element is eventually created to look similar to the one created with the first method. The only difference is that by drawing an AP Element, you can set the initial dimensions to anything you want instead of setting them at the default 200 pixels by 115 pixels. We'll use this method, in conjunction with the Grid (explained later in the chapter), to draw AP Elements when we build the tableless version of the Vecta Corp site.

Inserting Content into AP Elements

Inserting content into an AP Element is just as easy as it is to insert content into the cell of a table. To demonstrate this, I'll insert text into the first AP Element (the one at the top-left of the page) and insert an image into the second AP Element (the one on the center of the page that we just drew). To insert text into the first AP Element, place your cursor into the AP Element and start typing the text **Ada the Admin Assistant**. Now let's insert Ada's headshot photograph into the second AP Element. To do this, drag the `head_adatheadadminassistant.gif` image (located in the Images folder of our defined site in the Files panel) into the AP Element. When you've finished, the page will resemble Figure 7.4.

Notice that the text and the image are slightly undersized compared to the size of the AP Elements. To resize the AP Elements to correspond with their content, click, hold, and drag the resize handle located in the bottom-right of each AP Element to resize the AP Element to the size of the content within it. Finally, move the AP Element that contains the headshot photograph and place it just underneath the text AP Element. The result looks like Figure 7.5.

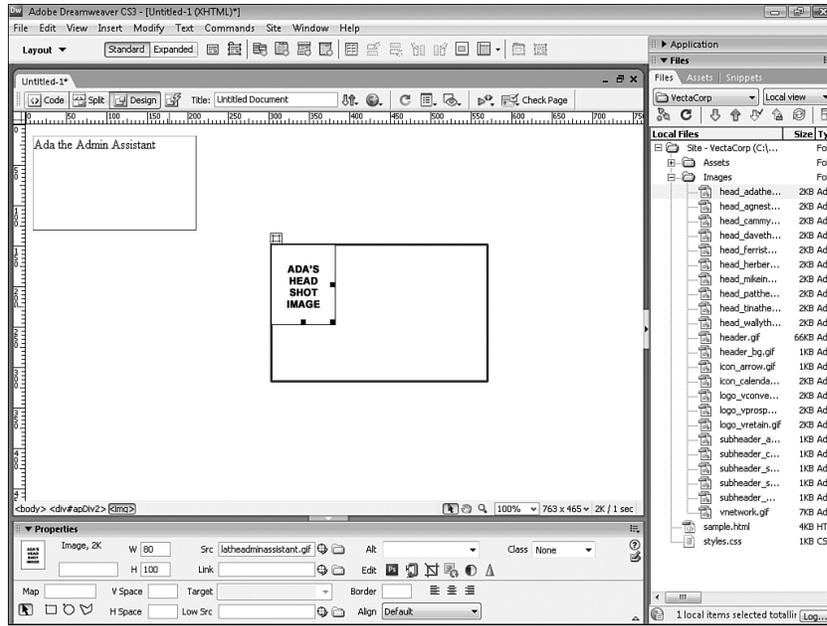


FIGURE 7.4 Add text to the first AP Element and drag an image into the second AP Element.

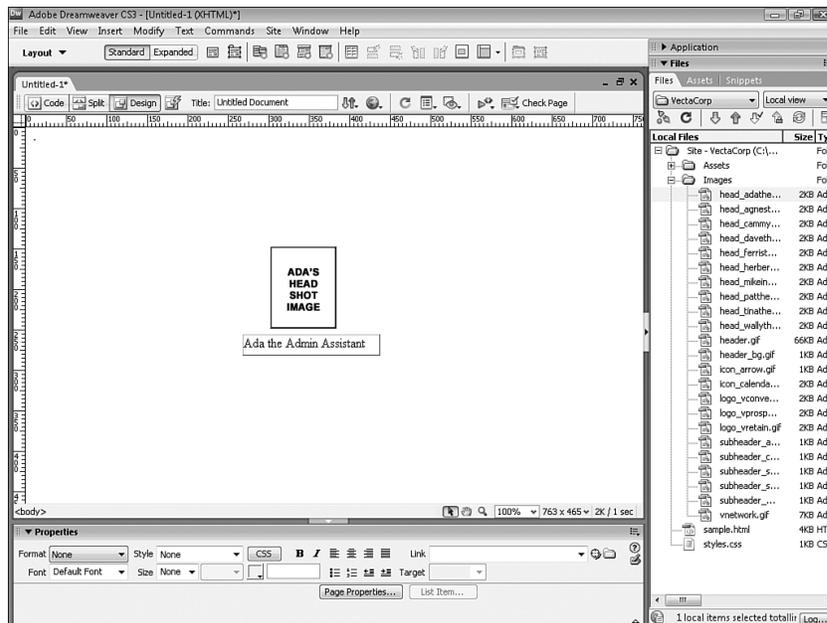


FIGURE 7.5 Resize the AP Elements so they match the size of the content within them and drag the headshot AP Element so that it sits just above the text AP Element.

You'll begin to notice the flexibility that AP Elements reveal: Adding them to the page is easy, moving them is easy, and resizing them relative to the content within them is easy. In the next few sections, we'll discuss the various properties you can set using the Property inspector and the AP Elements panel.

Modifying AP Element Properties with the Property Inspector

Like every feature in Dreamweaver, AP Elements have properties you can customize using the Property inspector. If you haven't noticed, the AP Elements Property inspector becomes available when you select an AP Element. As you can see in Figure 7.6, the Property inspector outlines properties that uniquely identify the AP Element, set the dimensions of the AP Element, set the stacking order or z-index of the AP Element, and so on. Also notice that moving your cursor just over the AP Element displays properties already set for the AP Element within a yellow ToolTip.

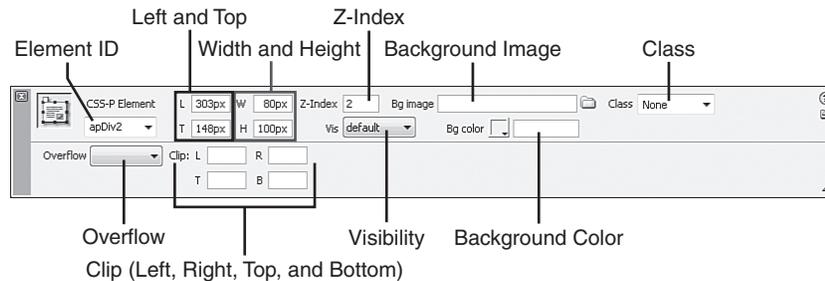


FIGURE 7.6 The AP Element-based Property inspector outlines numerous customizable properties.

A complete list of customizable properties outlined includes the following:

- ▶ **Element ID**—Add a value in this text box to uniquely identify the selected AP Element. By default, Dreamweaver adds the value `apDiv1`, `apDiv2`, and so on as you add AP Elements to the page. However, the default names mean little to us, so we'll change them. To demonstrate this, select each of the two AP Elements and name them **AdaImage** and **AdaText**, respectively. Also note that you can use only standard alphanumeric characters when naming an AP Element. Spaces, hyphens, slashes, and periods are not allowed.
- ▶ **L and T**—Enter values into these text boxes to set the position of the AP Element from the Left and Top of the browser. The Left and Top values are measured from the top-left point of the AP Element. If you're working with nested AP Elements, the Left and Top values are measured from top and leftmost corner of the parent AP Element.
- ▶ **W and H**—Enter values into these text boxes to set the width and height of the AP Element. If the content in the AP Element exceeds the width or height of the AP Element, Dreamweaver automatically stretches the AP Element to accommodate the content within it. As you'll see, you can change the Overflow property to set how the browser handles the excess content in the browser.

- ▶ **Z-Index**—Enter a value into this text box to set the stacking order of the AP Element. The lower the number, the lower the AP Element appears in the stacking order. The higher the number, the higher the AP Element appears in the stacking order. To demonstrate this property, select the AdaImage AP Element and change the Z-Index to **2**. Now select the AdaText AP Element and change that Z-Index to **1**. Position the AdaImage AP Element over the AdaText AP Element. As you can see from Figure 7.7, the AdaImage AP Element partially hides the AdaText AP Element. You know this is possible because the AdaImage AP Element has a higher Z-Index (2) than the AdaText AP Element (with a Z-Index value of 1).

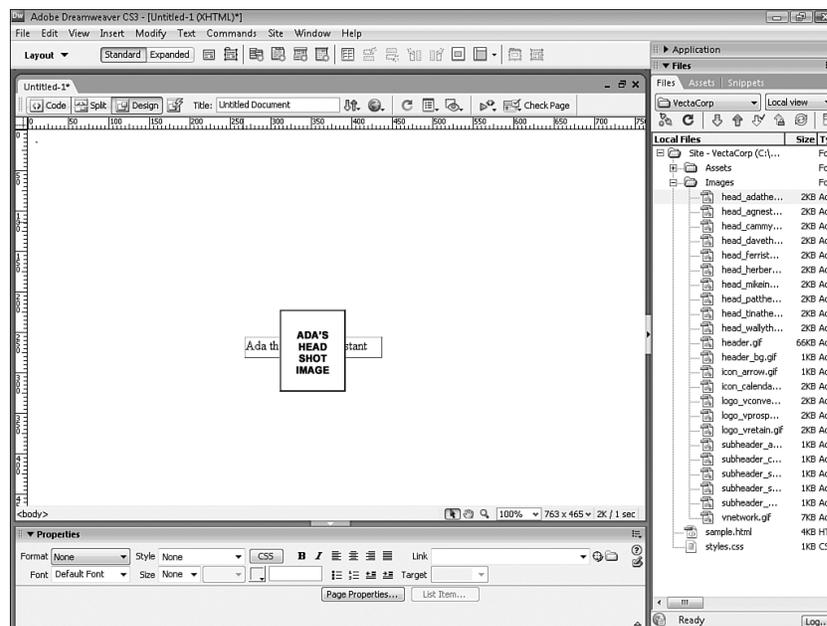


FIGURE 7.7 AP Elements with higher Z-Indexes appear above AP Elements with lower Z-Indexes.

- ▶ **Vis**—Select from the four options in this menu to set the visibility of the AP Element. Options include Default (which is essentially the same as Inherit), Visible (which makes the AP Element visible), Hidden (which hides the AP Element and the content within it), and Inherit (which assumes the visibility property of the parent AP Element when the AP Element is nested). If an AP Element is not nested within another AP Element, the default of Visible kicks in. When an AP Element is hidden, you'll want to use a scripting language such as JavaScript (the Show/Hide AP Element behavior in Dreamweaver's case) to dynamically set the visibility property of the AP Element.
- ▶ **Bg image**—Use the Browse to Folder icon to find and select an image to use as the AP Element's background. Background images in AP Elements are treated the same

as the page and tables, in that images smaller than the size of the AP Element end up tiling. However, you can always use the Repeat property in CSS to set how the background image for an AP Element tiles or repeats.

- ▶ **Bg color**—Choose a color from this color picker to set the background color of the AP Element. Leave this option blank to make the AP Element transparent.
- ▶ **Class**—When defining custom class selectors in CSS, pick a class from this menu to change the style of the AP Element according to the properties set out within the class rule.
- ▶ **Overflow**—Select an option from this menu to set how the browser should treat an AP Element when the content exceeds the width or height of the AP Element. Options include Visible, Hidden, Scroll, and Auto. Choosing Visible forces the browser to stretch the AP Element to accommodate the content in the AP Element. Choosing Hidden forces the browser to automatically crop the excess content when it exceeds the width or height of the AP Element. Choosing Scroll forces the browser to automatically add a scrollbar at the right and bottom of the AP Element. These scrollbars appear even if the content in the AP Element doesn't exceed the width or height of the AP Element. Choosing Auto forces the browser to automatically add a scrollbar at the right of the AP Element only if the content exceeds the height of the AP Element and to add a scrollbar to the bottom of the AP Element only if the content exceeds the width of the AP Element.
- ▶ **Clip**—Enter values into these text boxes to define the visible area of the AP Element. You can specify values for the Left, Top, Right, and Bottom coordinates to “clip” off content within the AP Element in the same way you'd crop an image in Fireworks or Photoshop.

As you'll see throughout the chapter, an AP Element is nothing more than a `<div>` or `` tag complete with a unique ID and a document-wide style rule (by default). Modifying properties in the Property inspector changes the document-wide style rule accordingly. To demonstrate this, switch to Code view. As you can see from Figure 7.8, each AP Element is nothing more than a `<div>` tag with an ID and a document-wide style rule.

You're probably thinking that document-wide styles aren't the best way of manipulating properties for AP Elements. If that's what you're thinking, you're right! Assuming that you were developing your site entirely in AP Elements, changing all the AP Element's properties in your site would be a nightmare if all the styles were developed within individual pages. A much better alternative is to create an ID within a style sheet that defines AP Element properties globally. When you want to make property adjustments (such as positioning properties), you make them once in the style sheet and instantly those changes are propagated to all of the pages that make use of the style properties. We'll demonstrate this process later in the chapter.

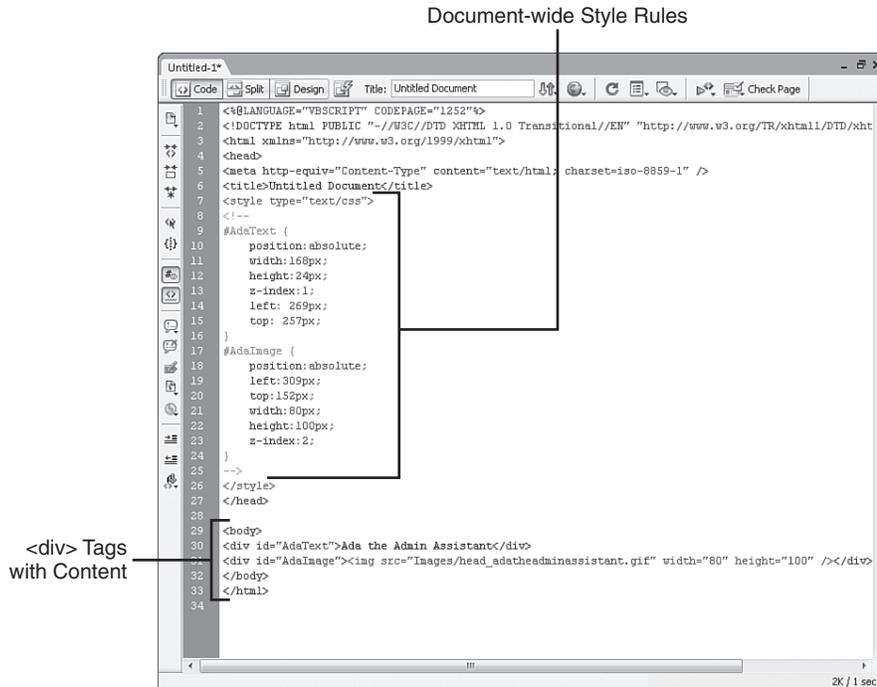


FIGURE 7.8 AP Elements are nothing more than a or tag complete with a unique ID and an inline style rule.

Modifying AP Element Properties with the AP Elements Panel

Another option for customizing AP Element properties is the AP Elements panel. You can use the AP Elements panel as a way to set an AP Element's ID, Visibility properties, Z-Index, and whether AP Elements can overlap one another. The AP Elements panel, shown in Figure 7.9, can be selected directly from the CSS Styles panel group or by choosing the AP Elements option from the Window menu or by pressing F2. Alternatively, you can right-click (Control+click) an AP Element's border to access the context menu and choose the AP Elements Panel option to open the AP Elements panel.

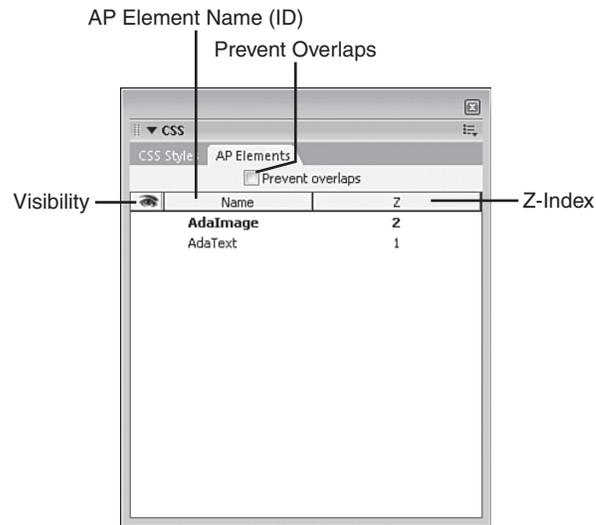


FIGURE 7.9 Use the AP Elements panel to visually set visibility, Z-Index, ID, and overlapping properties.

With the AP Elements panel open, notice that the style and overall appearance is strikingly familiar, looking much like the Layers panel does in an image-editing program such as Fireworks or Photoshop. For instance, you can drag AP Elements above or below other AP Elements, effectively changing the Z-Index of the AP Elements on the page.

TIP

If you drag an AP Element in the Document window that doesn't contain a Z-Index, Dreamweaver automatically adds one for you.

You can also click in the eyeball column to change the initial Visibility property for an AP Element. Finally, you can double-click in the Name column to change the ID of the selected AP Element. All this and more is possible through the AP Elements panel. The following customizable properties are revealed within the AP Elements panel:

- ▶ **Visibility**—You can click in this column for a particular AP Element to change the Visibility property. Icons include a closed eye (which represents Hidden), an open eye (which represents Visible), and no eye, the default (which represents Inherit).
- ▶ **Name**—You can double-click the AP Element name in this column to change the ID for the specific AP Element.
- ▶ **Z-Index**—You can double-click the AP Element in this column to change the Z-Index (stacking order) for the specific AP Element. You can also drag and drop AP Elements above or below one another to accomplish the same task.

- ▶ **Prevent overlaps**—When designing AP Element-based sites, click this button to prevent AP Elements from overlapping each other. This is important when you want to work with two AP Elements next to each other but want to prevent them from falling into an area used by another AP Element. We'll discuss this option further in the next section.

It's important to understand the place of the AP Elements panel. Sure, most of these properties can be set by selecting the AP Element and making the necessary change in the Property inspector. The difference between the Property inspector and the AP Elements panel, however, is that the AP Elements panel allows you to work with AP Elements even when they're hidden from view. Because the Property inspector becomes available only when an AP Element is selected, there is no way of changing the Visibility of an AP Element back to Visible after it's been Hidden without the use of the AP Elements panel. Also, it's important to note that the AP Elements panel represents the collection of AP Elements on the page, in contrast to the single AP Element that is covered by the Property inspector. The Prevent Overlaps check box prevents all AP Elements on the page from overlapping each other (discussed in the next section).

Preventing AP Elements from Overlapping

As mentioned in the previous section, you can use the Prevent Overlaps check box in the AP Elements panel to prevent all AP Elements on your page from accidentally overlapping each other when you position them. To demonstrate this feature, follow these steps:

1. Click the Prevent Overlaps check box in the AP Elements panel.
2. Try to drag the AdaImage AP Element above the AdaText AP Element as you did earlier in the chapter. You'll quickly notice that Dreamweaver prevents you from doing that.
3. To disable the feature, uncheck the Prevent Overlaps check box.

Remember, this feature comes in handy when designing AP Element-based websites because it can prevent you from accidentally moving an AP Element above or below another AP Element.

Nesting AP Elements

Similar to the process of nesting tables within a cell of another table, AP Element nesting is the process of placing an AP Element within another AP Element. By default, Dreamweaver doesn't automatically nest AP Elements; instead, Dreamweaver allows you to enable this option in the AP Elements category of the Preferences window. To enable AP Element nesting in your site, follow these steps:

1. Select Edit, Preferences (Dreamweaver, Preferences), or press Ctrl+U (Command+U). The Preferences dialog appears.
2. Select the AP Elements category.

3. Enable the Nesting: Nest When Created Within an AP Div check box.
4. Click OK.

With that preference set, you can now create AP Elements within other AP Elements. For the most part, the process of nesting an AP Element within a second AP Element is as simple as drawing or inserting an AP Element within another AP Element. To do this, follow these steps:

1. Insert or draw a new AP Element on the page and name it **BgAPElement**.
2. Select the AP Element and set the background color to gray by selecting a gray tone from the Bg Color field in the Property inspector.
3. Create a nested element by inserting a new AP Element within the existing BgAPElement AP Element. Name the new AP Element **NestedAPElement**.
4. Insert Agnes's headshot image into the new NestedAPElement AP Element so the result resembles Figure 7.10.

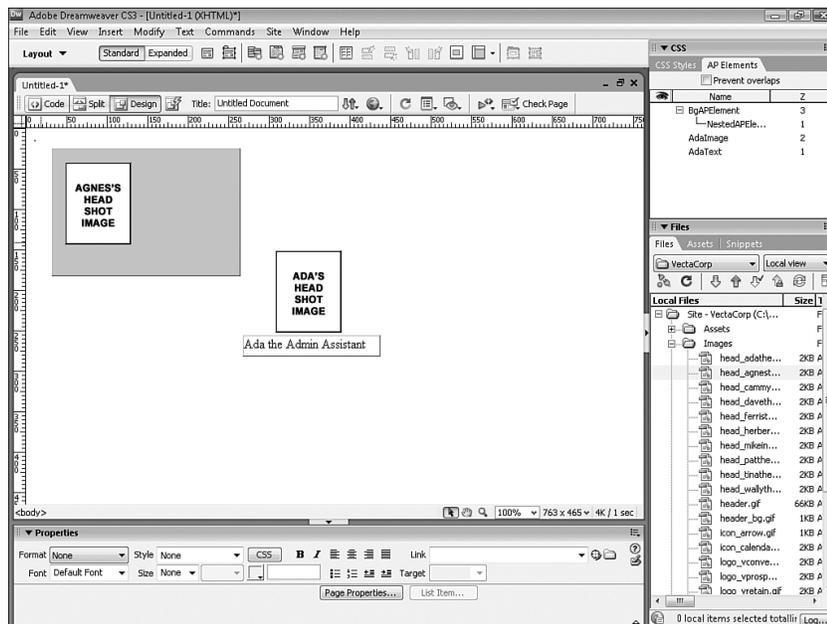


FIGURE 7.10 Nest an AP Element within an existing AP Element.

It won't be overly obvious that the new AP Element is nested within the BgAPElement AP Element. However, you can see that the AP Element is in fact nested by looking at the AP Elements panel, also shown in Figure 7.10. As you can see, the NestedAPElement AP Element appears as a child node beneath its parent BgAPElement AP Element. The real

benefit to nesting AP Elements becomes evident when you're working with Visibility properties. Because the default Visibility property of an AP Element is Inherit, the child AP Element always inherits the properties of its parent AP Element. What this means is that all we have to modify is the parent BgAPElement AP Element. Visibility changes made to this AP Element propagate down to all child AP Elements, effectively saving us from doubling our efforts.

Although it would seem logical that dragging a nested AP Element out of its container AP Element would unnest an AP Element, that isn't how you unnest elements. To unnest an AP Element, you must use the AP Elements panel to drag a child AP Element out of its current position and reposition it higher in the list in the AP Elements panel.

Deleting an AP Element

If you decide that a particular AP Element is no longer needed, you can easily delete it using one of two methods. The easiest way to delete an AP Element is to select it on the page and press the Delete or Backspace key on your keyboard. This action effectively removes the AP Element and its content from the document. The second option for deleting an AP Element from the Document window is to right-click (Control+click) the `<div>` tag in the Tag Selector and choose the Remove Tag option from the context menu. This action removes the `<div>` tag but does not remove the content from the Document window. The same holds true if you right-click the actual `<div>` tag on the page and choose Remove Tag `<div>`.

Designing Tableless Web Pages Using AP Elements

One of the biggest trends in web development is that of designing websites that don't use antiquated tables to control the structure of the site, but instead rely on AP Elements and CSS positioning properties to control the layout and flow of web pages. In the next few sections, you'll learn how Dreamweaver—in conjunction with the `<div>` tag and a little CSS—can create standards-compliant web designs.

NOTE

The term given to the process used for handling layout in CSS-based page designs is referred to as *CSS-P* (*P* for "positioning"). Technically, AP Elements in Dreamweaver use CSS-P for their positioning. In fact, if you select an AP Element in the Document window, you'll notice that label "CSS-P Element" is shown in the upper-left corner of the Property inspector.

Designing AP Elements Using a Style Sheet

In the previous sections of this chapter, you learned how easy it was to insert AP Elements using either the Insert, Layout Objects, AP Div command or the Draw AP Element option available from the Layout category in the Insert bar. Although these are

viable options for designing a site using AP Elements, they aren't the best choice. The reason for this is simple: When you use either of these methods to insert AP Elements on the page, Dreamweaver automatically assigns document-wide CSS positioning properties for each AP Element. As a result, the page becomes inflexible and nearly impossible to modify globally when numerous pages exist within the site.

A better alternative to inserting or drawing AP Elements onto the page is to create an external style sheet (which we already have for our project) and define numerous ID selectors that define the various sections of the page. For instance, we know our site will contain the following major sections based on the design we've been working with up to this point:

- ▶ **Header**—We know we'll have a header that resides near the top of the page. The header will have a width of 100% and a height of 227 pixels. We also know that the header will have a background image that spans the entire width of the browser. Finally, the header will contain the `header.gif` image that defines the logo and company name.
- ▶ **Navigation**—Just under the header but to the left of the page, we'll have a simple navigation menu. In this scenario, we can add an AP Element that will serve as a container for the five navigation links. Although the height doesn't matter in this case, we'll want to set the width at 200 pixels.
- ▶ **Content**—The third major section in the page is reserved for the content. This section will reside under the header but just to the right of the navigation area. Because our navigation area has a set width of 200 pixels and we want to keep the entire width of the page to a universally accessible 760 pixels, we'll set the content area's width to 560 pixels (200 pixels + 560 pixels = 760 pixels). Again, because a user will naturally scroll up and down the web page, the height for this section is irrelevant.

Now that we've outlined the major sections for our page, let's begin outlining the CSS rules. Because we know we'll have at least three major sections in the page (Header, Navigation, and Content), we can surmise that we'll need at least three CSS rules represented by three unique IDs titled `header`, `nav`, and `content`, respectively. To begin creating these selectors, follow these steps:

1. Create a new HTML page by choosing File, New. When the New Document dialog appears, select the HTML option from the Basic Page category, choose the `<none>` option from the Layout list, and click Create. Immediately save the page as `index.html`.
2. Attach the existing `styles.css` style sheet by opening the CSS Style panel, selecting the Attach Style Sheet icon, browsing for the `styles.css` file, and clicking OK.
3. Create a new CSS rule by clicking the New CSS Rule button in the CSS Styles panel. The New CSS Rule dialog appears.
4. Choose the Selector Type: Advanced option and enter the name **#Header** into the

Selector text box. Remember that the pound symbol represents ID (thus, the ID selector name is *Header*). When you finish making your modifications, the dialog should resemble Figure 7.11.

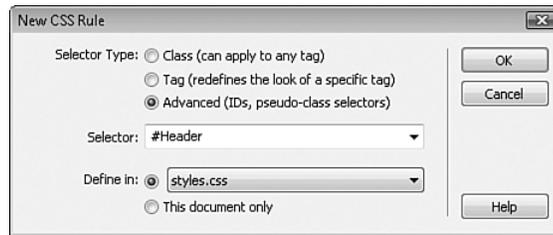


FIGURE 7.11 Create a new selector ID called Header.

5. Click OK. You're now ready to start defining the properties that will make up the Header rule and ultimately define the Header AP Element at the top of the page.
6. Switch over to the Positioning category. Set the property for Type to absolute, the width to 100%, the height to 227 pixels, the Visibility to Visible, and Top and Left Placement properties to 0. When you finish, the Positioning screen resembles Figure 7.12.

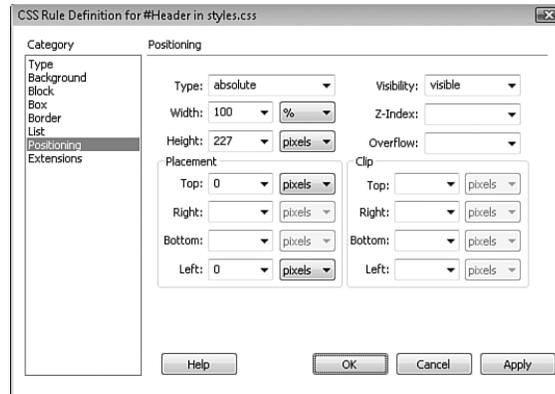


FIGURE 7.12 Modify the Positioning attributes for the Header rule.

7. Set the background image that will appear within the Header AP Element. You can do this by switching to the Background category. Browse to the header_bg.gif file from the Background Image field. Finally, choose the Repeat-x option from the Repeat menu to guarantee that the background image will tile only horizontally and never vertically.
8. Click OK. The new ID appears in the CSS Styles panel and styles.css is automatically opened in a new tab behind your current document.

That's it! You're now ready to define the style rules for the remaining AP Elements. To define the Nav rule, follow these steps:

1. Create a new CSS rule by clicking the New CSS Rule button in the CSS Styles panel. The New CSS Rule dialog appears.
2. Choose Selector Type: Advanced, if necessary, and enter the name **#Nav** into the Selector text box.
3. Click OK. You're now ready to start defining the properties that will make up the Nav rule and ultimately define the navigation AP Element.
4. Switch over to the Positioning category. Set the property for Type to absolute, the Width to 190 pixels (we'll add a padding of 10 pixels to the left of this AP Element shortly), the Visibility to visible, the Top Placement property to 227, and the Left Placement property to 0. When you finish, the screen should resemble Figure 7.13.

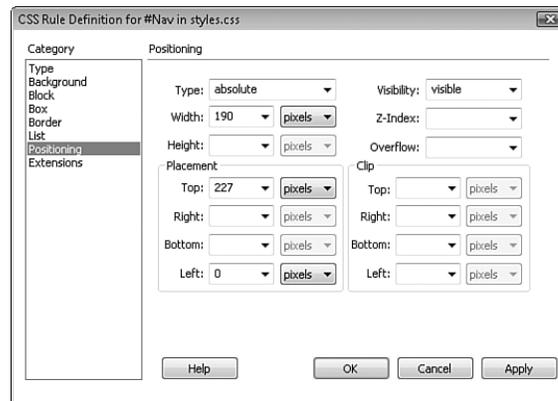


FIGURE 7.13 Modify the Positioning attributes for the Nav rule.

5. So that our navigation items don't run up against the left edge of the browser window, let's add some padding to this style rule. To do this, switch to the Box category. Disable the Same For All check box in the Padding category and enter a value of 10 pixels for the Left property.
6. Click OK. The new ID appears in the CSS Styles panel.

To finish up the design, let's define the Content style rule:

1. Create a new CSS rule by clicking the New CSS Rule icon in the CSS Styles panel. The New CSS Rule dialog appears.
2. Choose Selector Type: Advanced, if necessary, and enter the name **#Content** into the Selector text box.

3. Click OK. You're now ready to start defining the properties that will make up the Content rule and ultimately define the content AP Element.
4. Switch over to the Positioning category. Set the property for Type to absolute, the Width to 560 pixels, the Visibility to visible, the Top Placement property to 227, and the Left Placement property to 200.
5. Click OK. The new ID appears within the CSS Styles panel.

Now that we've outlined the properties for the three major sections of the site, we're ready to insert the tags that will act as containers for the content. The properties we've outlined will serve as the formatting and positioning attributes for the `<div>` tags we'll add next.

Inserting DIV Tags

Because we've already outlined the style rules using ID selectors in an external style sheet, we need only to insert three `<div>` tags into the page with IDs that match the ID selector names we created. To demonstrate this, let's insert a `<div>` for the Header portion of our page. You can do this by choosing Insert, Layout Objects, Div Tag. Alternatively, you could also click the Insert Div Tag icon (the first icon to the right of the Standard and Expanded buttons) in the Layout category of the Insert bar. Either method you choose produces the same result; the Insert Div Tag dialog launches and appears similar to Figure 7.14.

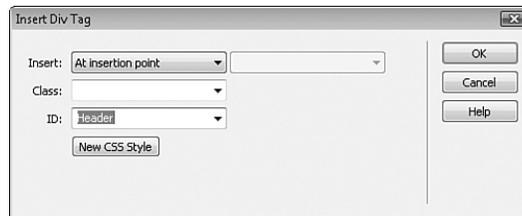


FIGURE 7.14 Use the Insert Div Tag dialog to create a new `<div>` tag in the page.

Although the dialog allows you to enter a class name and pick an insertion point for the tag, all we care about is entering the ID name that represents the ID selector we've defined within our style sheet. Because we're creating the header of the page, select the Header option from the ID drop-down menu (see Figure 7.14) and click OK. The AP Element is created, complete with the dimensions, placement properties, and the background image we defined.

By default, Dreamweaver adds the text *Content for id "Header" Goes Here* within the AP Element. Select this text and delete it so that we can add the `header.gif` image into the AP Element. When you've deleted the text, drag the `header.gif` image, located in the Images folder, into the AP Element. The result of the design resembles Figure 7.15.

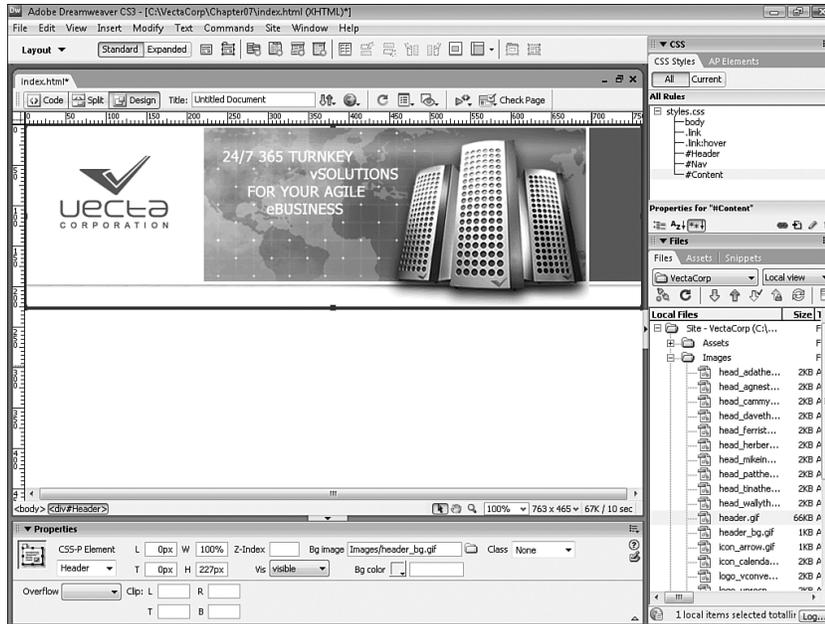


FIGURE 7.15 Add the header .gif image into the Header AP Element.

Now you're ready to create the second `<div>` tag that will define the navigation section. To do this, click the Insert Div Tag icon located in the Layout category of the Insert bar. When the Insert Div Tag dialog appears, select the Nav option from the ID drop-down menu and click OK. As you'll see, the Nav AP Element is added just below the Header AP Element.

Again, remove the *Content for id "Nav" Goes Here* text so that we can begin to define the navigational elements that will appear within this AP Element. When the placeholder text is deleted, add the first navigation element, Home, to the AP Element. Do this by following these steps:

1. Add the `icon_arrow.gif` image to the AP Element.
2. Select the image, add a V Space value of **2**, and choose the Absolute Middle option from the Align menu in the Property inspector. Setting these two properties gives you some spacing between arrow images vertically in the AP Element. Also, by setting the alignment to Absolute Middle, we'll be able to add our Home link just to the right of the image so that it lines up to the absolute center of the arrow image.
3. Select the arrow image, copy it, place your cursor next to the image, press Shift+Enter (Shift+Return) to create a line break, and then choose Edit, Paste. Repeat this process a few more times until you have a total of five arrows spanning vertically down the AP Element.
4. Insert the text **Home** just to the right of the first arrow image.

- Highlight the text and type the link **index.html** into the Link text box in the Property inspector. Also choose the Link option from the Style menu in the Property inspector to associate the correct style rule for the navigation link.

Now you can add the rest of the navigation elements: About Us (aboutus.html), Solutions (solutions.html), Support (support.html), and Contact Us (contactus.html). When you finish adding the navigation items, the result appears similar to Figure 7.16.

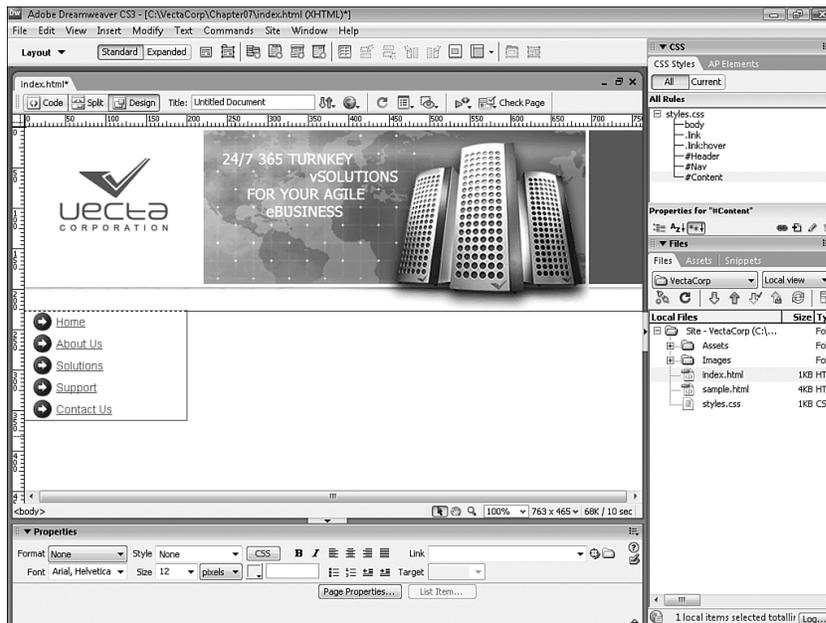


FIGURE 7.16 Add the rest of the navigational items to the Nav AP Element.

Finally, let's add the content section. Again, you can accomplish this task by clicking the Insert Div Tag icon in the Insert bar. When the Insert Div Tag dialog appears, choose the After Tag option from the Insert menu, choose the `<div id="Nav">` option from the menu that becomes enabled (this is done purely as a means of forcing Dreamweaver to organize the code properly), and choose the Content option from the ID drop-down menu. Click OK. The new AP Element is added below the Header AP Element but just to the right of the Nav AP Element. With the AP Element firmly in place, you can add the content that will fill that area. For instance, you might want to add the `subheader_welcome.gif` image as the subheader within the AP Element. You might also decide to add the text contained in the `home.txt` file in the Assets folder. When you've finished, the result should closely resemble Figure 7.17.



FIGURE 7.17 Add text and images to the new Content AP Element.

With your design complete, test the page in a browser by pressing F12 (Option+F12). As you'll see, the design renders without problems (refer back to Figure 7.1).

Of course, our design efforts don't have to stop here. We might want to create another AP Element with associated properties for our company events. Furthermore, we might want to add yet another AP Element with associated properties for the footer of the page. As you can begin to see, the design and implementation is simple.

Even better than the implementation is what has been added to the HTML. Switch to Code view and notice that the code is much cleaner and is minimized to roughly 30 lines of code (mostly text for the Content AP Element) in contrast to the 100 or so lines associated with the design when we were working with tables.

Converting AP Elements to Tables for Backward Compatibility

In the previous sections, we used AP Elements to create a standards-compliant, tableless web page. Although the design was simple to create, it doesn't do us or our users much good if they're using older browser versions such as Netscape 4. As you saw back in Figure 7.2, the page doesn't look very good in Netscape 4.

To solve this problem, we have to re-create the site using a backward-compatible model in tables. Before you become frustrated at the thought of having to rebuild the page using tables, know that there's an easy and automated way of accomplishing this task in the Convert AP Divs to Tables feature. Using this feature in conjunction with the Check Browser behavior, we can guarantee that our users will see the appropriately formatted

page regardless of what browser version they are using. To run through this process, choose Modify, Convert, AP Divs to Table. The Convert AP Divs to Table dialog appears, similar to Figure 7.18.

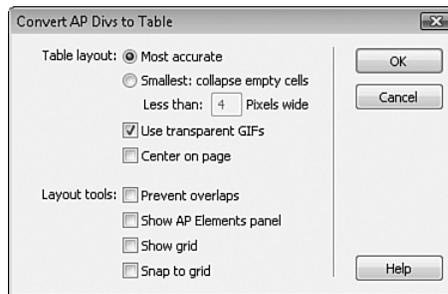


FIGURE 7.18 The Convert AP Divs to Table dialog facilitates the process of converting our CSS-based design to one that uses tables.

The following functionality is revealed in the Convert AP Divs to Table dialog:

- ▶ **Table Layout: Most Accurate**—Clicking this option button creates a cell for every AP Element, plus an additional cell for any spacing required to fill out the area.
- ▶ **Table Layout: Smallest: Collapse Empty Cells**—Clicking this option button specifies that the AP Elements' edges should be aligned if they are positioned within the specified number of pixels. We'll choose this option and leave the pixel width at the default of 4.
- ▶ **Table Layout: Use Transparent GIFs**—Checking this box ensures that the last row of the table has a 1 pixel by 1 pixel transparent GIF added to its cell. This is especially useful for guaranteeing consistent widths across all browser versions. For our purposes, uncheck this option now.
- ▶ **Table Layout: Center on Page**—Click this box to center the table on the page. Leave this option unchecked.

The rest of the features in this dialog exist for the reverse process (that is, converting tables to AP Divs). In this scenario, these check boxes merely provide viewable options such as the AP Elements panel, the grid, snapping, and so on. When you finish making the necessary adjustments, click OK. Immediately, our AP Elements-based page is converted to a table with a few minor discrepancies. For instance, it seems as though we lost our background image for the header. This is largely because the background image is attached to the ID selector in the style sheet. Dreamweaver has no way of knowing which cell in our new table to apply the background image to. To add it back in, select the header row within the table and use the point-to-file icon in the Bg field to browse to the header_bg.gif image. You may also need to merge the two cells that were created within the header row into one cell. You can do this by selecting both cells and choosing Modify,

Table, Merge Cells. Finally, you'll need to position the border between the navigation cell and the content cell so that the navigation and content cells are closer together. After you've made those modifications, save your work as `index_netscape.html`. Now try testing the page in an older version of Netscape.

TIP

If you don't have an older version of Netscape, you can download all archived versions at the following URL: browser.netscape.com/downloads/archive/.

As you can see from Figure 7.19, the page looks as it did in newer browser versions using AP Elements.



FIGURE 7.19 The page using tables looks as good in Netscape 4.7 as it did in newer browser versions using AP Elements.

NOTE

Although Dreamweaver does its best to convert your CSS-based design to tables, it's certainly not perfect. In many cases, additional tweaking of your design is mandatory to get the specific layout to look the way you want it to look in older browsers such as Netscape 4.

The last thing you'll want to do is add the Check Browser behavior to the `index.html` page. This behavior allows us to create functionality so that when the user visits the `index.html` page using a Netscape 4 browser, the user is automatically redirected to `index_netscape.html`. However, if users visit the `index.html` page using a newer browser, they stay right where they are, on the CSS-based `index.html` page. To add this behavior, follow these steps:

1. Open `index.html`.
2. Open the Behaviors panel by choosing the Behaviors option from the Window menu.
3. Choose the `<body>` tag from the Tag Selector and choose the Check Browser option located within the ~Deprecated submenu in the Add Behavior list within the Behaviors panel. The Check Browser dialog appears.
4. In the dialog, enter the value **4.7** (the version that you should be using) into the Netscape Navigator Version text box. Also select the Stay on This Page option from the Otherwise menu. Because Internet Explorer 4 can display the AP Elements page just fine, select the Stay on This Page option from both Internet Explorer menus. Also select the Stay on This Page option from the Other Browser menu. Finally, type the `index_netscape.html` path in the URL text box.
5. Because most other browsers can also display AP Elements, choose the Stay on This Page option from the Other browsers menu as well. When you finish, the Check Browser dialog resembles Figure 7.20.
6. Click OK.

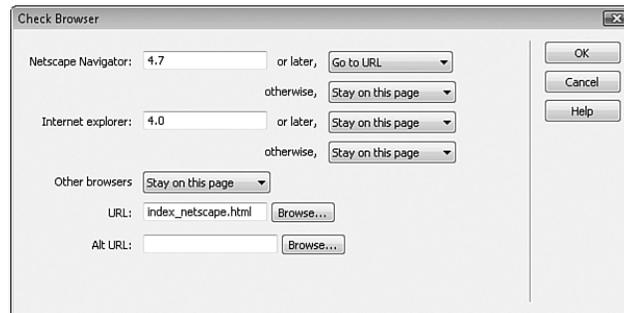


FIGURE 7.20 Configure the Check Browser behavior so that users visiting your site using Netscape 4 browsers are automatically redirected to the correct page.

With the CSS-based page still open, try to preview your page in Netscape 4. Notice that you are automatically redirected to the `index_netscape.html` page.

NOTE

Obviously, if this were a full site with numerous pages, you'd want to convert all of the pages and links within the site.

Built-In CSS Page Layouts

One of the hottest web-development techniques is the process of laying out web page structures using CSS instead of traditional HTML-based methods such as tables or frames. As the Web evolves, so too does the process of creating web pages. Therefore, tables are viewed as an archaic method of page structuring. As a web developer you should be aware of that fact and begin learning and integrating standards-compliant methods into your site designs. In the previous few sections we began to do just that. As you might recall, you looked at ways of building web pages that rely on CSS for the positioning and structuring of web pages. What little HTML we used was meant purely for integrating CSS rules or for adding imagery and media elements to your pages. As you might also recall, the learning curve for creating standards-compliant CSS page designs is relatively higher as compared to tables or frames. To help you along, Dreamweaver CS3 integrates a collection of carefully crafted and comment-rich CSS Page Layouts.

This section will get you up to speed with Dreamweaver's CSS page layouts. As you'll see, working with CSS-based structuring techniques has never been easier—thanks to this new collection of page layouts. To build a new web page based on prestructured CSS-based layout, start by selecting File, New. The New Document dialog appears. Select the HTML option from the Blank Page category. Up to this point in the book we've usually selected the <none> option from the Layout list and clicked Create. This time, however, take a closer look at the additional options within the list. As you can see from Figure 7.21, additional prebuilt CSS-structured options exist.

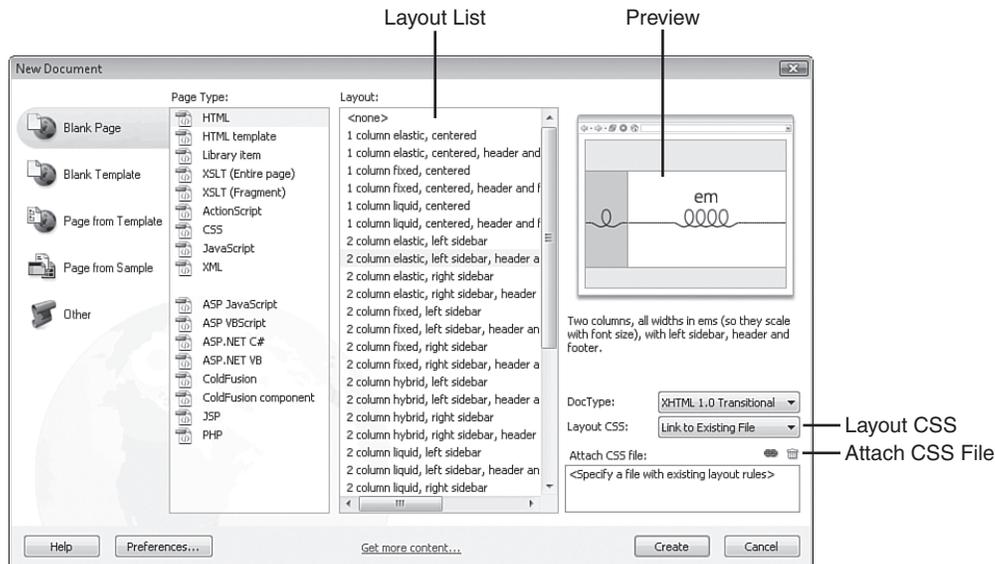


FIGURE 7.21 Choose from a list of prebuilt CSS-structured page layouts.

Numerous options exist within this dialog for creating and handling the CSS that results, including the following:

- ▶ **Layout**—The Layout list contains 32 prebuilt CSS-structured layouts for you to use. For our purposes, select the 2 column elastic, left sidebar, header and footer option.
- ▶ **Preview**—The previous window displays an iconic representation of the selection you make from the Layout list. This preview is a rough approximation of what your design will be structured around.
- ▶ **Layout CSS**—Choose an option from this menu to instruct Dreamweaver as to how to handle the CSS that will accompany your layout. Options include Add to Head (adds all the CSS to the head of the page creating a document-wide style sheet), Create New File (adds all the CSS to a new external style sheet file), and Link to Existing File (adds all the CSS to an existing external style sheet file). When you select the Link to Existing File option, you'll be required to browse to the existing CSS style sheet using the Attach Style Sheet button available from the Attach CSS file group of options. For our purposes, choose the Add to Head option.
- ▶ **Attach CSS file**—When you choose the Link to Existing File option from the Layout CSS menu, this group of options becomes immediately available. You can choose the Attach Style Sheet icon to browse to and attach an existing style sheet. Alternatively, you can delete a style sheet from this list by clicking the Remove Style Sheet icon.

Now that we have a layout selected and we've instructed Dreamweaver as to where it should place the accompanying CSS, click Create. Immediately, a new Document window is created, complete with your selected layout, similar to Figure 7.22.

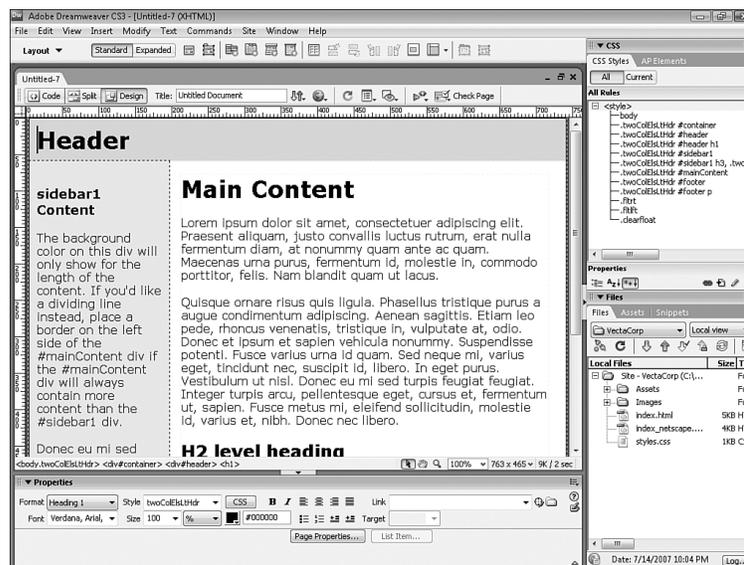
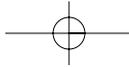


FIGURE 7.22 A new document window instance is created complete with your selected layout.



At a glance, you can see that the content is placed within DIV tags. And, like you saw in the previous section, the positioning and structure of those DIV tags are controlled using a series of carefully crafted classes, also visible in the CSS panel in Figure 7.22. Now try switching to Code view. Immediately you'll notice how many comments have been added to guide you through the relatively new process of using CSS to structure web pages. Now comes the fun part—removing the default text that appears in the DIV tags and replacing it with the Vecta Corp content. See if you can accomplish this task on your own!

Summary

As you've seen in this chapter, AP Elements offer a clean, standards-compliant alternative to developing websites. Using CSS in conjunction with `<div>` tags affords you the capability to cleanly separate positioning and formatting properties from structural code.

In this chapter, you learned about the various methods for inserting AP Elements, modifying properties outlined by AP Elements, and structuring pages using AP Elements and CSS positioning properties. Furthermore, you looked at some of the new Dreamweaver CS3 CSS-based page layouts and how building CSS-based designs is not only easy, but also effortless when you take advantage of them.

In the next few chapters, we'll begin to shy away from CSS and take a much different direction by beginning to discuss important concepts as they relate to team collaboration and task automation.

