

## APPENDIX D

# Working with Frames and Framesets

In the mid-1990s, Netscape introduced a proprietary HTML extension known as *frames*. The idea was that web pages at the time used up far too much memory and required excessive bandwidth to be viewed within a time span that didn't bore the user. On top of that, Netscape hoped that framed sites could be viewed only within Netscape, further positioning its browser implementation as the standard of the time.

Designers of the time fell right in line with the concept, using framed websites to present consistent unmoving navigation bars that remained static even when content within a separate frame was scrolled. Even better, users visiting sites using 14.4Kbps or 28.8Kbps modems enjoyed the fact that although framed sites had an initial load time, they ran much smoother and loaded quicker during the viewing experience.

Although frames definitely offer another page structuring option for the web designer, disadvantages have emerged. This chapter aims at demystifying framed sites, discussing the advantages, disadvantages, and even how to build them within Dreamweaver.

## Understanding Framed Websites

Framed websites are generally easy to spot; chances are you've probably used one and may or may not have known it. A framed website generally consists of numerous parts, starting with the frameset. As you've seen up to this point, browsers typically open one page at a time. When a site uses frames, this concept is no different. However, rather than opening numerous files at one time as you might

### IN THIS APPENDIX

- ▶ Understanding Framed Websites
- ▶ Working with Frames and Framesets
- ▶ Adjusting Frame Attributes
- ▶ Adjusting Frameset Attributes
- ▶ Adding Content to Frames
- ▶ Saving Frames and Framesets
- ▶ Targeting Frames
- ▶ IFrames
- ▶ Targeting Browsers That Don't Support Frames

think might occur with framed sites, a browser generally opens a *frameset*. The frameset is usually split into numerous frames that make up the content of the site within the frameset. Figure D.1 illustrates a framed website in use.

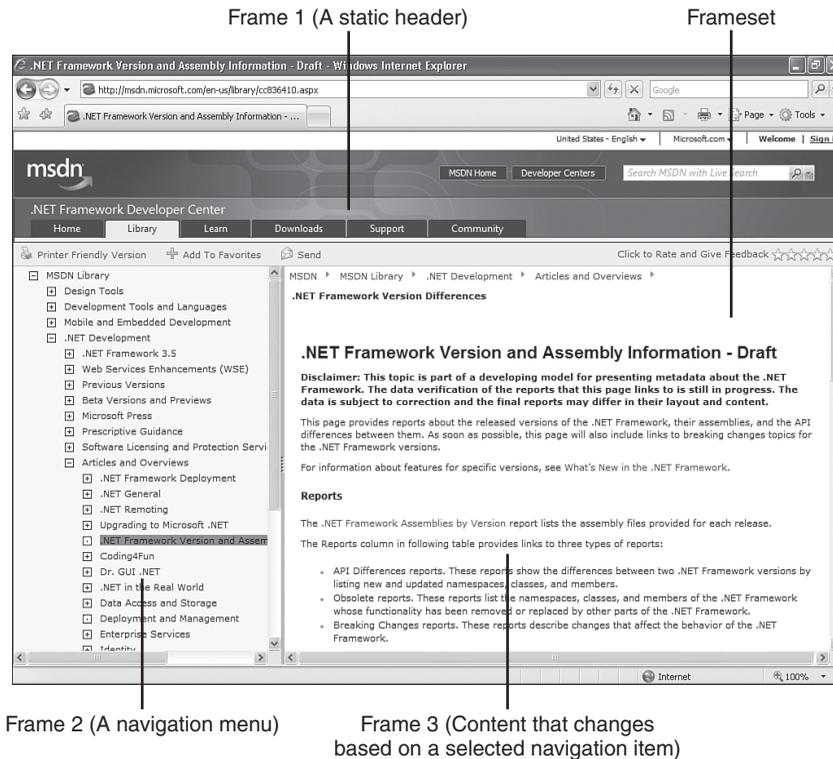


FIGURE D.1 The MSDN Library website is a good example of a framed website.

Microsoft's MSDN Library website is a good example of a framed website. From the callouts in Figure D.1, you can clearly see that the page is split into three frames, each making up important parts of the frameset. The frame near the top contains important top-level navigation items that relate directly to Microsoft and the major sections within the overall site. The frame on the left makes up the navigational structure for the MSDN Library specifically. Finally, the frame on the right makes up the content that appears when a link within the navigational tree on the left is clicked. Navigation items in the top and left frames always remain static, even when content within the right frame is scrolled up and down or left to right. Even better, the navigational structure represented in the top and left frames loads once and only once. Only the content in the right frame is loaded every time a navigation item within the left tree is clicked, essentially improving load time and the overall user experience. Furthermore, the frame that makes up the navigation menu on the left can be manually resized by the user. Shown in Figure D.2, resizing the frame allows for easier reading of the navigation menu.

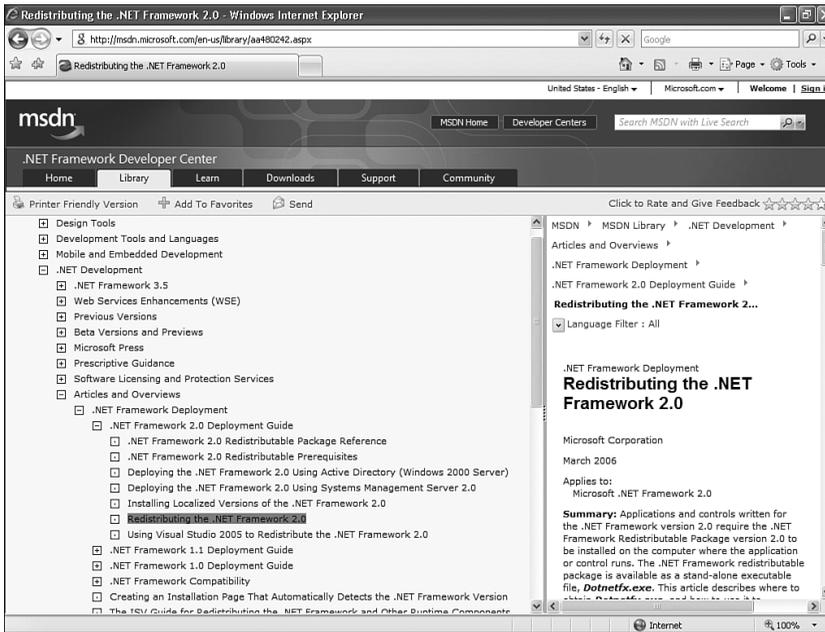


FIGURE D.2 You can resize the frame to make the navigation tree easier to read.

Although this example represents a classic and advantageous use of a framed website, it has disadvantages. Let's dissect the advantages and disadvantages next.

## The Advantages of Frames

The advantages of setting up a website that uses frames can be profound. As you can see from the MSDN Library website, all elements that are constant throughout the web experience, such as the navigation menu at the top and left, do not have to be reloaded every time a link is clicked. This navigation structure provides advantages for both author and user. From the author's viewpoint, if a navigation bar needs to be changed, perhaps to add or remove a menu item, only one document has to be amended. If the navigation bars were integrated into each page (as would be likely if the page had been developed using tables or CSS), then every site document would have to be changed (unless you were using library items or templates). From a user's standpoint, framed navigation menus result in much "lighter" documents compared to pages with static navigation menus because the pages don't contain an excess of HTML required to set up the layouts. This can also be advantageous for users with dial-up modems, which according to the latest statistics, is still a fairly high number of Internet users. By having the navigation menu reside within a separate frame and new content appear within a larger, autonomous frame, the site can appear more complete and will load quicker per navigation click. Accessing new documents and swapping them out of the same location saves download time and offers an easier navigation experience for the web user.

In addition to the benefits presented thus far, the site is somewhat configurable. As you saw in Figure D.2, the navigation frame can be resized by the user. Again, this is advantageous to both the developer and the user. From a developer's standpoint, there are no concerns with respect to limiting the size of the links to fit within the frame. From a user's standpoint, if the content within the navigation frame exceeds the size of the frame, you merely resize the border to expand the viewable area of that frame.

Although the advantages presented would be enough to convince anyone to want to use frames, there are numerous disadvantages. In fact, as you'll see, the disadvantages overshadow these few advantages.

## The Disadvantages of Frames

Along with all the advantages frames can provide, they also offer numerous disadvantages to the user and several challenges for a web designer. The most important of these challenges is search engine placement. Search engines use web crawlers and robots to index pages on websites. *Crawlers* are programs that wander from site to site, gathering information on web pages and returning that information to the search engine for indexing. These programs often don't index frameset documents because framesets contain no content, only instructions to create frames, so there's little or nothing to index. *Robots*, however, index documents with content, and when links to these documents appear in search results, the search engine user is taken directly to the page, bypassing the frameset and ultimately the content (including navigation) that was meant to be seen.

Another disadvantage is bookmarks. Browsers bookmark only the frameset document on a framed website, not the pages showing in the frames. If you navigate through the MSDN Library site and try to bookmark particular pages, the bookmark will point only to the relevant frameset and not to the page itself, so you'll never be able to bookmark the page you want unless you open it in a fresh browser window using the right-click (Control-click) context menu. To fix this problem, the MSDN website exposes a link within the page itself. The Add to Favorites link shown in Figure D.3 points to JavaScript code that allows the user to bookmark the frameset with the selected page.

Yet another disadvantage when working with framed websites is printing. This issue stems from the same problem exposed by bookmarks in that, when you go to print a page, the browser has no idea what page within the frameset you are referring to. Ultimately, what prints out is a blank page that represents the frameset and not the intended page. To be sure that you print the document contained within a frame, you really need to open the document in a new browser window, away from the frameset, and print from there. From a developer standpoint, you can add a Print This Page link (similar to the one shown in Figure D.3) that makes use of JavaScript code to print out the correct page (frame).

Furthermore, framed websites present problems for the visually impaired. Traditional screen readers lack the capability to distinguish between the frames within a frameset and end up locking a significant number of users out of the website unless a "no frames" alternative is available (discussed later). Modern screen readers, such as JAWS for Windows or VoiceOver for Mac, are more sophisticated and can handle frames. However, you shouldn't assume that every visually impaired user makes use of a modern screen reader.

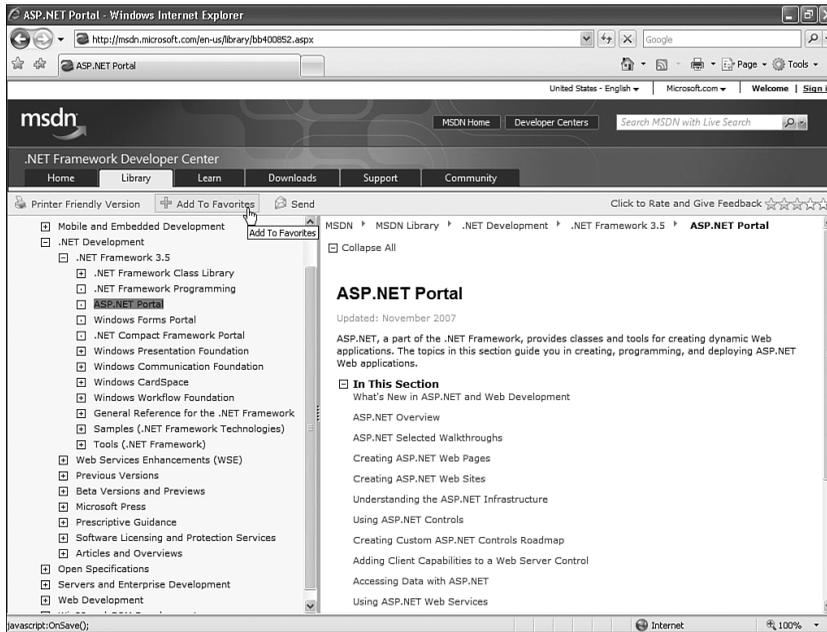


FIGURE D.3 Sites get around the bookmark issue in frames by creating their own link so that users can manually add the bookmark to their favorites.

Finally, framed websites are just no longer cool. Trends have long passed by working with framed sites, opting for the more standards-compliant CSS alternative. Although framed websites still have their place, they shouldn't be used as a method for creating an entire website. In most cases, I'll only use frames when a client either requires it or I'm building an application similar to the MSDN Library, where positioning a static tree-like navigation menu makes more sense within a frame than with a table's cell or CSS `<div>` tag. However, these scenarios are rare.

## Working with Frames and Framesets

Now that you have an idea about how frame-based websites work and are aware of their advantages and disadvantages, let's begin working with them within Dreamweaver.

Like the rest of the chapters within this book, you can work with the examples in this appendix by downloading the files from the book's website. You'll want to replace the files for your defined Vecta Corp site (C:\VectaCorp) with the files located in `\Exercises\AppendixD\`.

Dreamweaver makes working with framed websites a snap. Initially, everything begins with the Frames panel, which is available by choosing Window, Frames. The Frames panel allows you to easily select and work with frames and framesets.

As you might have noticed, the Frames panel looks fairly useless. It's not until you've actually created a new frameset document that the panel comes alive. Let's do that now.

## Creating a New Website Using Prebuilt Framesets

The quickest and easiest way to create a new framed website is to use one of the prebuilt framesets available from the New Document dialog box. To create a new frameset using this method, follow these instructions:

1. Choose File, New. The New Document dialog box appears.
2. Choose the Page from Sample section on the left and then choose Frameset from the Sample Folder category.
3. Pick a frameset type from the Sample Page category. Although 15 prebuilt framesets exist, the most popular by far is the Fixed Top, Nested Left option, shown in Figure D.4. Select it.

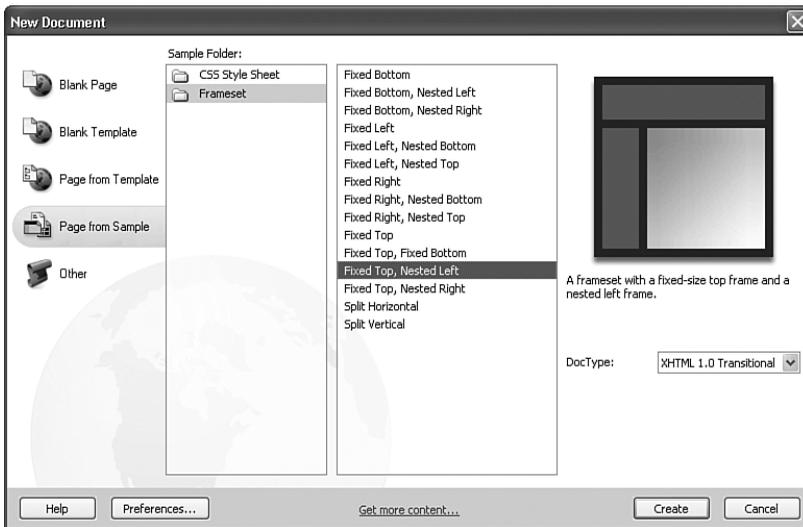


FIGURE D.4 The most popular frameset is the Fixed Top, Nested Left option.

4. Click Create. The new frameset appears within the Document window, split into the three frames shown in the preview icon in the New Document dialog box. You'll also notice that the Frames panel, shown in Figure D.5, visually displays the frameset as it appears within the Document window.

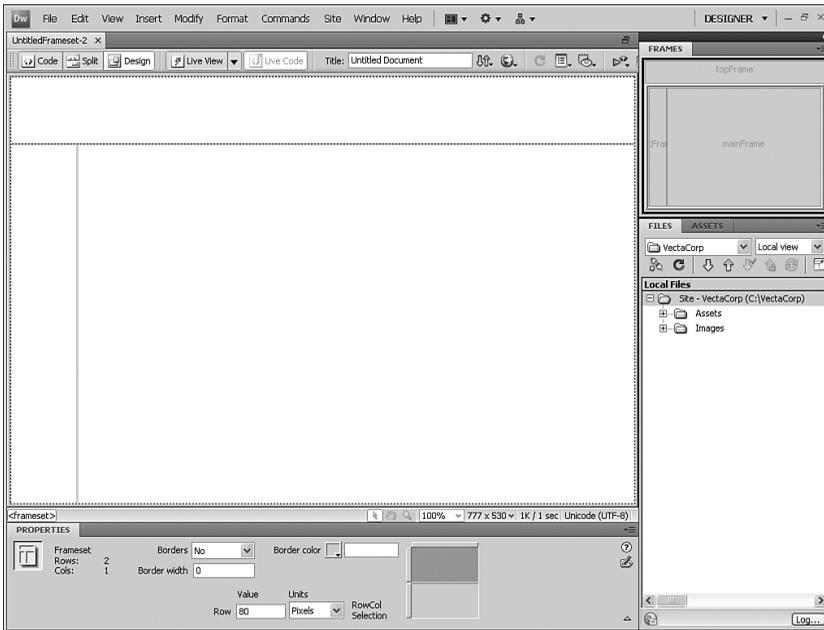


FIGURE D.5 The Frames panel displays a visual representation of the design within the Document window.

## NOTE

After you click Create, by default Dreamweaver displays accessibility options as they relate to frames and framesets. Remember, if you don't find yourself working with Accessibility features at this point, this dialog box can be turned off by navigating to the Preferences dialog box (available from the Edit menu), choosing the Accessibility category, and unchecking the Frames check box. If you didn't see this dialog box, it's because you've already configured these preferences earlier in the book.

Although this is certainly not the only way of creating framed websites, it is by far the simplest. The next section explores alternative methods of creating framesets.

## Alternate Methods of Creating Framesets

Although using one of Dreamweaver's prebuilt frameset files is certainly easy enough, it's hard to ignore three other methods for generating framesets within a web page. Regardless of which method you choose, you can create a similar frameset design.

The first alternative method involves options from the Insert menu. To see this method in action, follow these steps:

1. Create a new blank HTML document by selecting File, New. Choose the Blank Page section followed by HTML from the Page Type list, select <none> from the Layout category, and then click Create. A new blank HTML page appears.
2. Select Insert, HTML, Frames, Top Nested Left.

The result of this command is similar to the Fixed Top, Nested Left option available from the New Document dialog box. In fact, the Frames menu contains 13 options similar to those found within the list of prebuilt framesets in the New Document dialog box.

The second method for generating a frameset document involves the Modify menu. To use this method, follow these steps:

1. Create a new blank HTML document by selecting File, New. Choose HTML from the Page Type list (available by choosing the Blank Page category), ensure that <none> is selected in the Layout category, and click Create. A new blank HTML page appears.
2. Place your cursor within the page and select Modify, Frameset, Split Frame Up.
3. Place your cursor in the bottom frame and choose Modify, Frameset, Split Frame Left.

This design is similar to the previous options covered thus far.

Finally, you can create the same frameset structure by choosing an option from the Frames icon located within the Layout category within the Insert panel. As you can see from Figure D.6, most of the options available from the list of prebuilt frameset files are available.

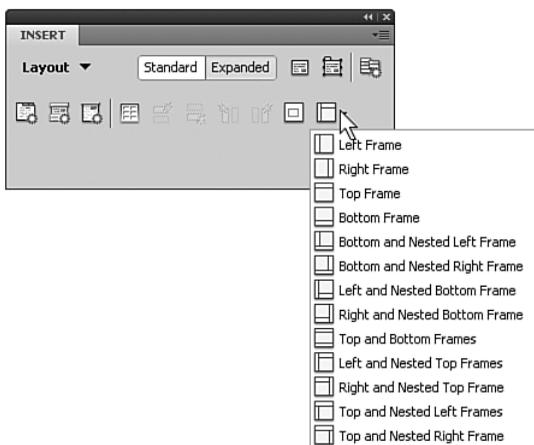


FIGURE D.6 Use the Frames icon within the Layout category in the Insert panel to pick from a list of prebuilt frameset files.

Now that you have a frameset (one which contains frames) created, the question becomes, how do you go about deleting frames within the frameset? Although deleting a frame from a frameset isn't as intuitive as clicking a button or choosing an option from a menu,

it's still fairly simple. To delete a frame from within a frameset, click and drag the frame border of the unwanted frame off the page. The frame is removed from the frameset.

### TIP

If you can't see the frame borders, you can enable them by selecting the Frame Borders option from the Visual Aids menu in the Document toolbar.

To prepare for the rest of the examples in this appendix, close all Document window instances except for the first one you created using the New Document dialog box method.

## Adjusting Frame Attributes

With the frameset now created and open, you're ready to begin adjusting attributes for the three frames within the frameset. Many of the attributes associated with frames exist so that you can set frame names, which ultimately determine the target of the frame, whether content within a frame should scroll, whether frame borders appear, and so on. Here is a complete list of frame properties outlined within the Property inspector for frames:

- ▶ Frame name
- ▶ Frame source
- ▶ Frame scroll
- ▶ Frame resize
- ▶ Borders and border color
- ▶ Margin width and height

You use the Property inspector to modify these frame properties. To see the Frames Property inspector, select one of the frames within the Frames panel. As you can see from the callouts in Figure D.7, the Frames Property inspector includes the modifiable properties just listed and described in the following sections.

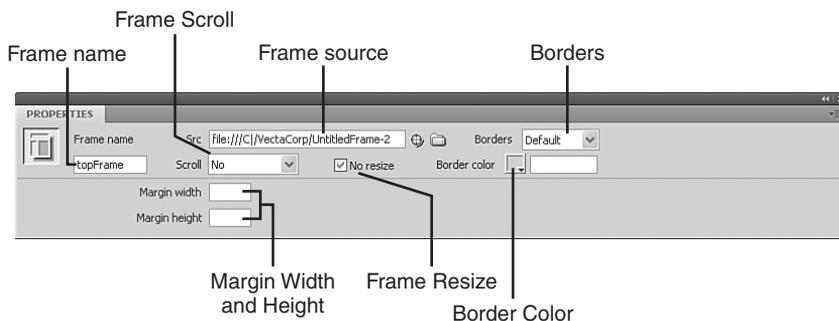


FIGURE D.7 The Frames Property inspector becomes available when you are selecting a frame from the Frames panel.

## Changing Page Properties

Changing page properties for a frame is no different from changing the properties for an individual page, as we've done throughout the book thus far. It's important to remember that frames are individual HTML files wrapped together using a frameset. To modify the page properties for a frame, simply place your cursor within the frame and select the Page Properties option from the Modify menu. Let's do it together:

1. Place your cursor within the top frame (don't select it from the Frames panel—just place your cursor within the frame in the Document window) and choose Modify, Page Properties. The Page Properties dialog box appears.
2. Choose the Appearance (HTML) category and then set the background image by browsing to the file `header_bg.gif`, located within the Images folder of your project.
3. Click OK.

This action effectively creates a new document-wide style with one tag redefinition for the `<body>` tag in this frame. You're probably asking yourself, "Why didn't we modify the `<body>` tag redefinition in our external style sheet file with this background property and then just attach the style sheet?" Although this seems like a viable solution, think about the rest of the frames. Had we redefined the `<body>` tag in the external style sheet file to include this background image, all frames would have had their backgrounds set to the same image—not the effect we want, right? That's not to say that we can't use the `styles.css` file within our frames, though; we just don't want to set the `<body>` tag so that it includes a background image. We'll still link the style sheet in so that we can take advantage of the page margins and default font properties. Let's do that next. To attach the style sheet to this frame, follow these steps:

1. Open the CSS Styles panel; if it's not open, choose Window, CSS Styles.
2. Click the Attach Style Sheet icon in the CSS Styles panel. This is the first icon from the left in the bottom-right icon group in the CSS Styles panel. The Attach External Style Sheet dialog box appears.
3. Browse to your `styles.css` file located within the root of your project.
4. Make sure the Link option is selected in the Add As category. Leave the Media menu unchanged.
5. Click OK.

Notice that the external style sheet now appears in the CSS Styles panel below the document-wide style sheet. Also notice that we've redefined two `<body>` tags. This is cascading at its finest. Although we've redefined the `<body>` tag within our external style sheet, we didn't set the background image there. That would give us unwanted results within every page that uses that external style sheet file. Instead, we redefined the `<body>` tag again using a document-wide style. This style, combined with the style in the external style sheet file, gives us a combination of properties we can be happy with for this frame only. Now place your cursor in the left navigation frame and repeat the process of attaching the style sheet. Finally, place your cursor in the right content frame and attach the style sheet one more time. Remember, these are three different files—therefore the style sheet must be imported three times. After you've done that, you're done! You'll notice that all page

margins are now set, the default font and size are set, and the top frame even has its background image set independently of the other frames. When you're finished, the frameset should resemble Figure D.8.

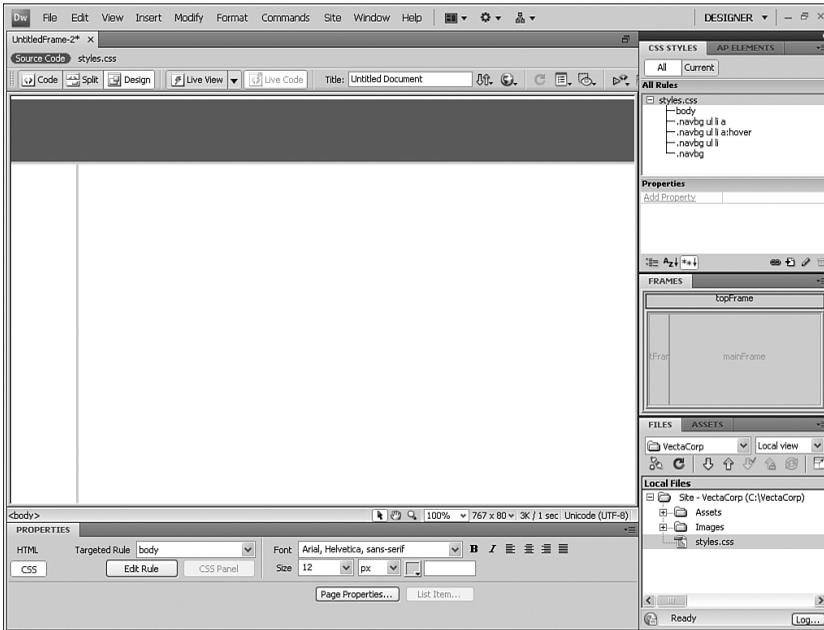


FIGURE D.8 Set the background image of the top header frame and attach the style sheet to every frame individually.

## Setting Frame Names

The next property you'll want to set is the frame name. A required property, the frame name is important because it creates a new target for links in other frames. In Chapter 2, "Building a Web Page," we discussed four targets: `_blank`, `_parent`, `_self`, and `_top`. When working with frames however, frame names also count as targets. This means that if you create three frame names (such as `header`, `nav`, and `content`), you can effectively establish links in the `nav` frame that target (open their contents in) the `content` frame. Creating a frame name is easy, and this is where the Frames panel really comes into play. To name a frame, follow these instructions:

1. Select the top frame currently labeled `topFrame` in the Frames panel. The Property inspector changes to accommodate the editing of frame properties.
2. In the Frame Name text box, enter the text **Header**.
3. Press Enter or Tab to commit the change.

That was easy, right? As you'll notice, the new name is given to the frame and is visible in the Frames panel. Now perform the same steps for the left frame, naming it **Nav**, and the

right frame, naming it **Content**. When you're finished, your Frames panel should resemble mine (see Figure D.9).

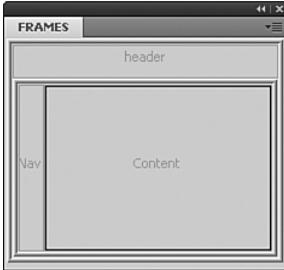


FIGURE D.9 Name the three frames Header, Nav, and Content, respectively.

## Setting Scroll Properties

The next property you'll want to set is the Scroll property. Represented by the Scroll menu in the Property inspector, you can set this property to enable or disable the appearance of scrollbars within a frame. The following options are available within this menu:

- ▶ **Yes**—Enables scrolling for a frame. When this option is selected, a scrollbar always appears within the frame, even if the content does not exceed the size of the frame.
- ▶ **No**—Disables scrolling for a frame. When this option is selected, a scrollbar never appears within the frame, even if the content exceeds the size of the frame.
- ▶ **Auto**—Set the Scroll property to Auto when you want the browser to automatically show scrollbars only when the content exceeds the size of the frame.
- ▶ **Default**—Choosing Default excludes the Scroll property from appearing within code. Setting this option to Default doesn't set a value for the corresponding attribute, allowing each browser to use its default value, which is typically Auto.

Setting the scrollbars for a frame is as simple as selecting a frame from the Frames panel and choosing an option from the Scroll menu. For our project, choose No for the Header frame, No for the Nav frame, and Auto for the Content frame.

## Frame Resize

The next property you want to set is the frame No Resize property. Enabling this check box guarantees that users will not be able to grab the border (if it exists) and drag it to resize the frame. Set this property by selecting each frame in the Frames panel and

enabling the check box. For our example, make sure that the box is enabled for all three frames.

#### NOTE

Earlier I mentioned that the capability to resize frames was one of the advantages to using frames. Although this is certainly the case, we'll prevent our users from resizing our Vecta Corp frame because the navigation items within the Nav frame are short and do not warrant allowing the users to resize.

## Adjusting Borders and Border Color

Another useful set of properties you can set are the Borders and Border color properties. Use the options in the Borders menu in the Property inspector to enable or disable borders within a frame in the browser. The following options are listed within this menu:

- ▶ **Yes**—Choose this option to show borders for a frame.
- ▶ **No**—Choose this option to hide borders for a frame.
- ▶ **Default**—Choosing the Default option doesn't set a value for the corresponding attribute in code, allowing each browser to use its default value, which is typically Yes.

#### NOTE

It's important to note that a border is hidden only when all frames that share the border have Borders set to No, or when the frameset's Borders property is set to No and the frames sharing the border have Borders set to Default.

The Border color picker allows you to set a border color for all of the frame's borders. This color applies to all borders that touch the frame and overrides the specified border color of the frameset.

For our example, set the Borders property to No for all frames and don't pick a color from the color picker.

## Margin Width and Height

The final set of frame-based properties you can set in the Property inspector are the Margin Width and Margin Height options. These properties, represented by the two text boxes in the Property inspector, are specific only to Netscape 4 and are set rarely, if ever. If

you are targeting older Netscape browsers, though, know that these properties function similar to the margin properties in the Page Properties dialog box in that they allow you to specify how much of a margin width or height the frame should have in Netscape 4.

## Adjusting Frameset Attributes

Framesets, like frames, also have properties you can customize. Properties such as borders, border widths, border colors, row heights, and column widths can all be customized from the Frameset Property inspector. Here's a complete list of the properties outlined within the Frameset Property inspector:

- ▶ Borders
- ▶ Border width
- ▶ Border color
- ▶ Row height
- ▶ Column width

To access the Frameset Property inspector, select the frameset from either the Frames panel or the Tag Selector. To select the frameset from the Frames panel, click the outer border of the frame in the panel. For instance, I'll select the Fixed Top frameset by clicking the outer border of the frame, shown by the cursor placement in Figure D.10.

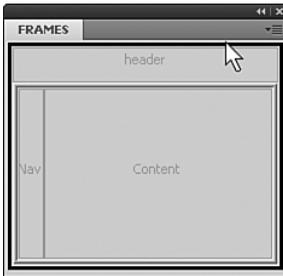


FIGURE D.10 Select the frameset by clicking the outer frame border in the Frames panel.

If you do it right, the Property inspector becomes the Frameset Property inspector. Alternatively, you can select the frameset by selecting a frame within the Frames panel. The Tag Selector shows the hierarchy of tags, including the `<frameset>` tag, that the selected frame belongs to. Choose the `<frameset>` tag from the Tag Selector, shown in Figure D.11, to select the frameset and have the Property inspector change to the Frameset Property inspector.

```
<frameset> <frameset> <frame#Content>
```

FIGURE D.11 Select the **<frameset>** tag within the Tag Selector to select the frameset.

Either method you choose enables the Frameset Property inspector for the particular frameset. Remember, the Fixed Top, Nested Left frameset is really a frameset within a frameset. This means that you'll have to access and modify two framesets instead of one.

After you've selected the frameset, the properties for the frameset are revealed within the Property inspector. These properties are also highlighted by the callouts in Figure D.12. Each of these properties is discussed in the following sections.

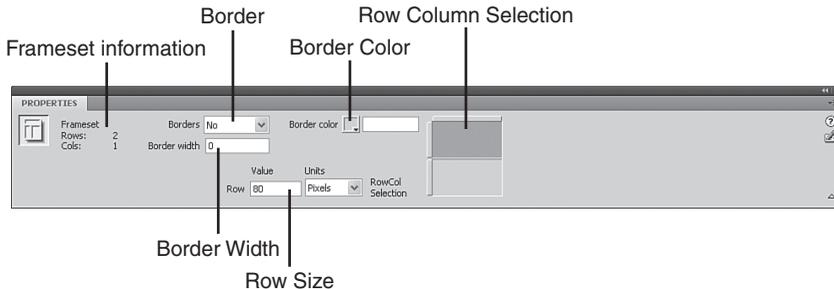


FIGURE D.12 The Frameset Property inspector exposes properties for working with the frameset as a whole.

## Borders, Border Width, and Border Color

As was the case with frames, framesets also support the modification of border, border width, and border color properties. Represented by the Borders menu, the Border width text box, and the Border color picker, you can set these properties in an effort to control how the frameset treats borders. Because the frameset is technically considered the parent of the frames it contains, properties set here propagate down to the frames contained within the frameset. For instance, if you choose Yes for Borders, a border width of 5, and a color of green, those changes are automatically set for the frames within the frameset. This means that as long as you set the Borders property to Default in the Frameset Property inspector, changes within the Frames Property inspector relating to borders can be left alone.

For our project, set the frameset's properties to **No** for Borders, **0** for Border width, and no color option within the Border color picker.

## Row Height and Column Width

Depending on the frameset you're working with in the Frames panel, the Frameset Property inspector will tailor itself to support changing the row height or the column width. For instance, selecting the outer edge of the frameset in the Frames panel reveals the Row Height property in the Property inspector; selecting the inner frameset reveals the Column Width property in the Property inspector.

**NOTE**

Remember, our frameset (which appears to be one page split into three frames) is actually one frameset (two frames) nested within a second frameset. The nested frameset contains two columns, and the outer frameset is split into two rows. For this reason, selecting the outer frameset reveals the Row Height property in the Property inspector, and selecting the inner frameset reveals the Column Width property in the Property inspector.

You can use these properties to accurately change the sizes of the columns or rows that ultimately govern the size of the frames in the frameset.

When making size modifications to frames, it's important to understand *how* they are sized. Generally, frames either have an absolute pixel size or a relative size. For example, if you had a frameset that consisted of a top and bottom frame, you could set the top frame to a specified pixel size. The bottom frame size would be relative to the size of the browser window or, to be more specific, 1. To see this in action, we need only to point to our frameset. You can see that switching between the top and bottom frames in the shaded frameset icon (Row Column selection) in the Property inspector gives you the two different sizes for the two frames in the outer frameset. The top frame is set to a pixel value while the bottom frame is set to a relative value of 1. The same holds true for the nested frameset. In this case, the left frame has a pixel value and the right frame has a relative value of 1. This arrangement allows the user to expand and contract the browser window without collapsing the sizes of the frames in their corresponding framesets. Now that you understand this concept, let's resize our frames.

The easiest way to change the size of a frame is to mouse over the border until your cursor changes to a two-way arrow, and then click and drag. The only problem with resizing frames like this is that both rows and columns are automatically converted into absolute sizes. To avoid setting frames to absolute sizes by accident, you can resize them by using the Row Height or Column Width text boxes in the Property inspector. To set the sizes for our project, follow these steps:

1. Highlight the outer frameset in the Frames panel.
2. The Property inspector should now show the Row Height text box. Enter a value of **227** pixels. This value corresponds to the size of the header image, which we'll add later. Press Enter to set the size. The row height for the frame should change.
3. Select the nested frameset in the Frames panel. You can make it easy on yourself by selecting a frame in the Frames panel and choosing the inner `<frameset>` tag in the Tag Selector. Either method causes the Property inspector to change to support the modification of the Column Width.
4. Enter a value of **217** and press Enter. This alteration should give us enough room to add a navigation bar and will coincide with the gray vertical bar outlined in the `header.gif` image.

Now that you understand how the attributes can be set for frames and framesets in the Property inspector, let's move on to adding content into the frames.

## Adding Content to Frames

Now that you have your frame structure in place and you've made the necessary modifications to attributes for every frame, you're ready to begin adding the content that will make up the shape of the site. Adding content to each frame is just as simple as adding content to an individual web page. For instance, let's begin by adding the `header.gif` image to the header frame. You can do this by dragging the image out of the Images folder (from within the Files panel) and dropping it onto the header frame. The result looks similar to Figure D.13.

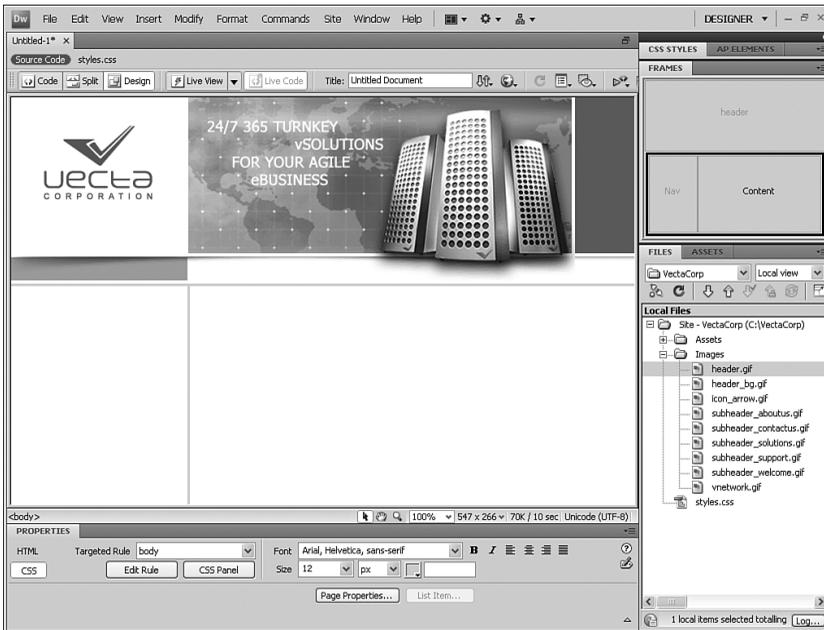


FIGURE D.13 Add the `header.gif` image to the header frame.

As you can see, the `header.gif` image in this project contains a gray area that will coincide with the navigation frame's background color. This color scheme will make the site's frames appear seamless. To add the background color and navigation menu to the nav frame, follow these steps:

1. Place your cursor in the nav frame and change the background color by choosing Modify, Page Properties. From the Page Properties dialog box choose the Appearance (HTML) category, and then click the Background color picker and sample the gray color from the vertical bar in the `header.gif` image. The hexadecimal value `#A5AEB5` should be created within the text box. Click OK to close the Page Properties dialog box.
2. With your cursor still in the nav frame, insert a new table by choosing Insert, Table. The Table dialog box appears.

3. Give your table **5** rows, **2** columns, a width of **175** pixels, a border of **0**, cell padding of **0**, and cell spacing of **3**. Click OK. The new table will appear within the frame.
4. Add the `icon_arrow.gif` image to the five cells in the first column.
5. Finish off the table by adding text that will eventually represent the links to our pages: **Home**, **About Us**, **Solutions**, **Support**, and **Contact Us**. Don't worry about linking them just yet.
6. Select the table and choose the Center option from the Align menu in the Property inspector. The completed navigation menu should resemble Figure D.14.

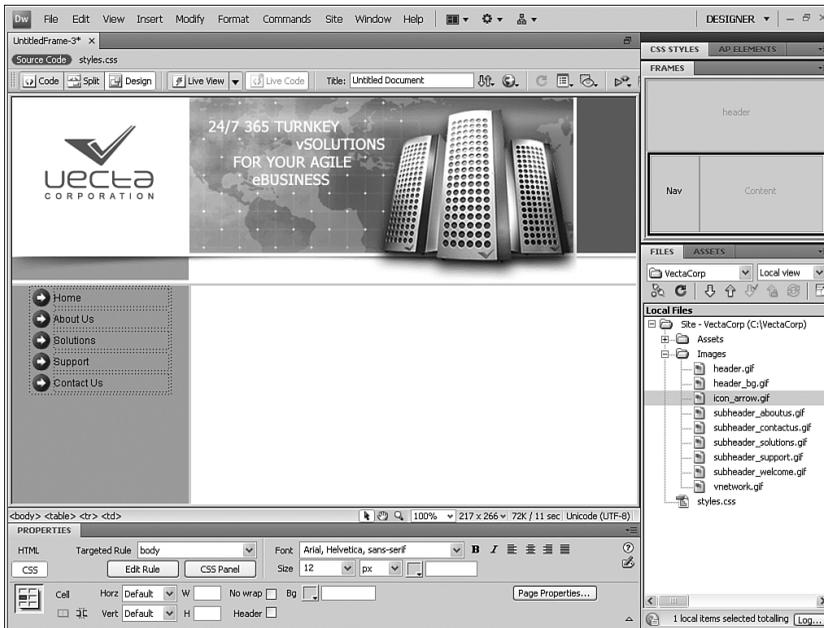


FIGURE D.14 Add the arrow icon and the five text elements that will eventually become links, and center the table within the frame.

You can finish off the design of the site by adding the final elements to the content frame. To do this, follow these steps:

1. Place your cursor in the content frame and insert a new table by choosing Insert, Table. The Table dialog box appears.
2. Give your table **3** rows, **1** column, a width of **470** pixels, a border of **0**, cell padding of **0**, and cell spacing of **5**. Click OK.

3. The new table is inserted into the frame. Now add the `subheader_welcome.gif` image located in the Images folder into the first cell.
4. Now add the welcome content into the third cell. You can do this by opening the `home.txt` file located in the Assets folder, selecting all the text, copying it, and then pasting it directly within the third cell. The final design should resemble Figure D.15.

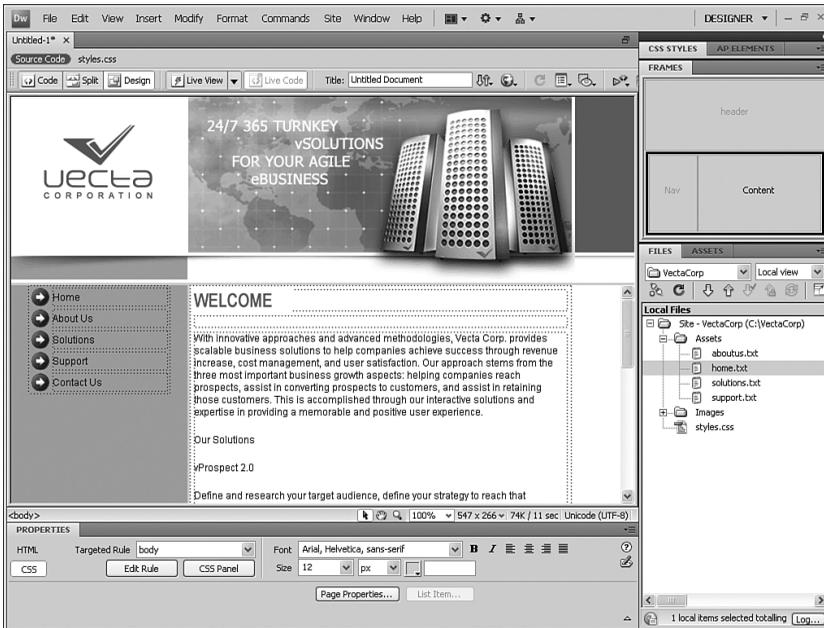


FIGURE D.15 The complete design should have a header in the header frame, a navigation menu in the nav frame, and content in the content frame.

Now that the content is firmly in place, you're ready to save the frames and the frameset. Let's do that next.

## Saving Frames and Framesets

Because our frameset appears to be complete, now is a good time to save it. Saving frames and framesets, however, isn't as straightforward as saving an individual page. With frames and framesets, you must first save all the individual frames and then save the frameset file. Let's do that next:

1. Place your cursor in the header frame and choose File, Save Frame As. Save this page as **header.html**.
2. Place your cursor in the nav frame and choose File, Save Frame As. This time, save the page as **nav.html**.
3. Place your cursor in the content frame and choose File, Save Frame As. Save this page as **home.html**.
4. Select the outer edge of the frameset from within the Frames panel to select the entire frameset. Now choose File, Save Frameset As. Save the frameset as **index.html**.

When you're finished, you'll end up with four HTML files in your Site Files list, similar to Figure D.16.

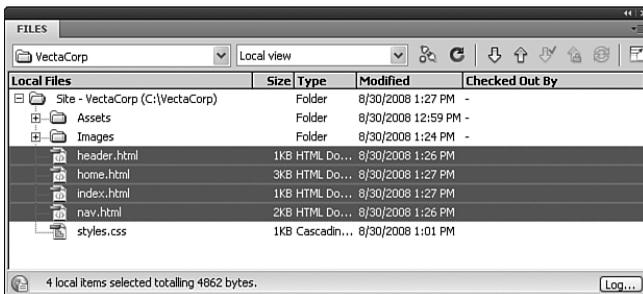


FIGURE D.16 After saving your frames and frameset, you should end up with four HTML files.

## NOTE

You might be wondering why we saved the content frame as `home.html`. Because this is the home page, saving this page as `home.html` makes more sense than saving it as a different name, such as `content.html`. People will link to `home.html`, `aboutus.html`, `solutions.html`, `support.html`, and `contactus.html`. Although the frame name *content* makes sense for the frame, the filename `content.html` doesn't apply in our navigational scheme box.

Finally, preview your site in the browser by choosing F12 (Option-F12). As you can see from Figure D.17, the site looks seamless. As a matter of fact, it doesn't really look like it was created using frames at all; instead, it looks like our table-based or CSS-based site.

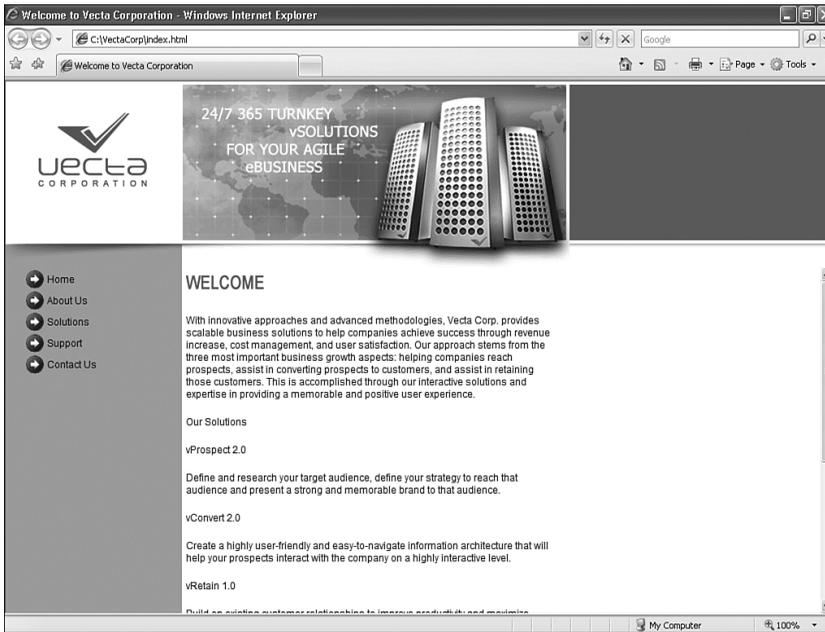


FIGURE D.17 The complete site in the browser.

## TIP

You might find yourself constantly making changes to numerous frames. Rather than saving each individual frame and frameset every time, you can choose the File, Save All command to save all open files that have been modified since the last save. You can also preview your page in the browser. Every time you select this option, Dreamweaver automatically saves all files that have been modified since the last save. However, this feature works only if the Preview Using Temporary File option is *not* enabled in the Preferences dialog box.

## Targeting Frames

Before we can call our framed website complete, we'll want to add links to the navigation menu. If you recall, frames are unusual in that navigation menus can link to content that opens in other frames. This means that your navigation menu can remain static and doesn't have to load every time—only the content being linked loads every time within an autonomous frame. This is possible because of *targets*. In Chapter 2, we discussed four targets: `_blank`, `_parent`, `_self`, and `_top`. When working with frames however, frame names also become targets. You can create hyperlinks in the nav frame that open

home.html in the content frame, for instance. To see this in action, let's create a simple link for the one page that we have complete: home.html:

1. Highlight the Home text in the nav frame.
2. Use the point-to-file icon located next to the Link text box in the Property inspector to point to the home.html file located within the Files panel.
3. Now we must set the target. To do that, select the Content option from the Target menu, as shown in Figure D.18. This option sets the content frame as the target in which the home.html file will open.

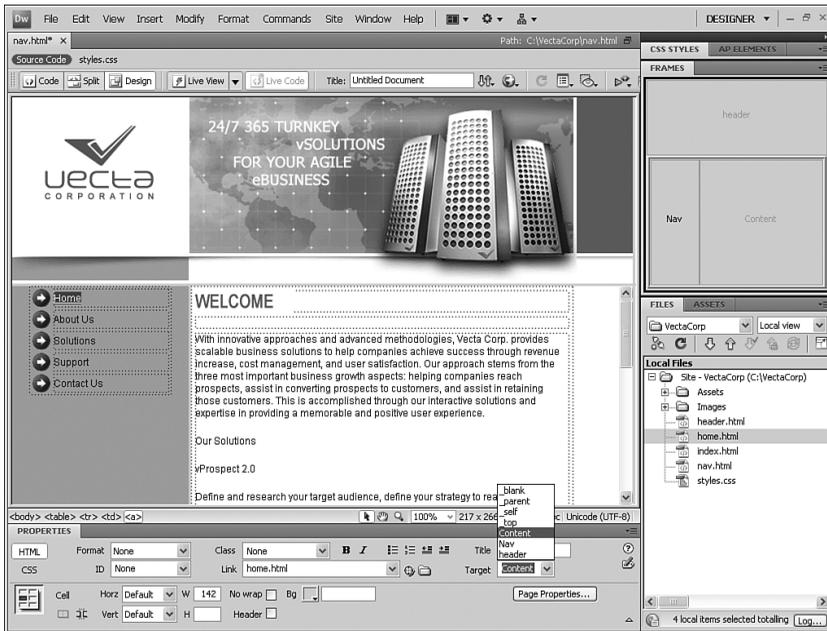


FIGURE D.18 Select the Content option from the Target menu in the Property inspector.

4. Had we not set the Content target, the browser would default to `_self`. What this means is that when the Home link was clicked, the home.html file would open within the nav frame. Not exactly the effect we're going for. To finish off the link, apply a style to it by selecting the `link` option from the Style menu in the Property inspector.

You've effectively created a link for the home.html file. To see the rest of the links in action, create some simple pages to link to. To do this, follow these steps:

1. Create four new blank HTML documents by selecting File, New. Choose HTML from the Page Type list (available by choosing the Blank Page category), check that `<none>` is selected in the Layout category, and click Create. A new blank HTML page appears. Repeat this process four times.

2. With five tabs available in the Document window (including our frameset), add a subheader to each page. For example, add the subheader\_aboutus.gif file to the Untitled-1 page, add the subheader\_solutions.gif file to the Untitled-2 page, and so on.
3. When you finish adding the headers, save each page with the same names as their link counterparts in the navigation menu. You should now have aboutus.html, solutions.html, support.html, and contactus.html.
4. Repeat steps 1–4 at the beginning of this section for each link in the navigation bar, using the point-to-file icon to associate each text element as a link with its corresponding file in the Files panel. Each time you create a target, make sure that you select the Content option from the Target menu in the Property inspector and that you attach the link style from the Style menu to each hyperlink. When you've finished, the navigation bar will look like Figure D.19.



FIGURE D.19 All links within the navigation bar should now be created.

Preview your page in the browser to see the result of the link target.

## IFrames

Inline Frames, often referred to as *IFrames* or *floating frames*, allow you to create freestanding, scrollable framed widgets anywhere on the page without the need or hassle of having to work with framesets or frames. To add and work with an IFrame, follow these steps:

1. Create a new blank HTML document by selecting File, New. Choose HTML from the Page Type list (available by choosing the Blank Page category), ensure that the Layout category is set to <none>, and click Create. A new blank HTML page appears. Save the page as `iframe.html`.
2. Choose Insert, HTML, Frames, IFrame. Dreamweaver will automatically switch to Split view and add the `<iframe></iframe>` tags in code.
3. The `<iframe>` tag has several attributes that you need to add to make it work correctly. The most important is the `src` attribute. The `src` attribute (short for *source*) specifies which document/URL you want to use in the IFrame. For demonstration purposes, point it to `http://www.dreamweaverunleashed.com`.
4. You'll also have to specify a unique name, a width, and a height. The formatted tag should look something like this:

```
<iframe src="http://www.dreamweaverunleashed.com"
name="myFrame" width="400" height="200">
</iframe>
```

5. Save your work and preview the page in the browser. The result will resemble Figure D.20.

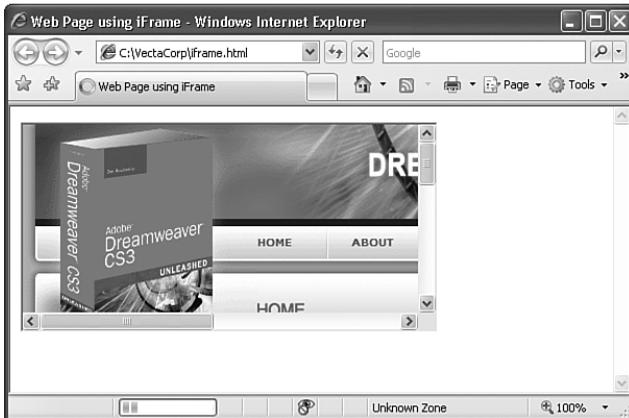


FIGURE D.20 IFrames allow you to create freestanding, scrollable panes anywhere on the page.

## Targeting Browsers That Don't Support Frames

As you can see, framed sites offer advantages in that they are quick to develop and are relatively fast loading for users with slower connections. However, what if a user browses to your framed site using a text-only browser (such as Lynx)? In a situation such as this, your site (and content) won't load for the user. To remedy this, you'll want to tweak your design slightly to accommodate these types of users. This can be accomplished easily in Dreamweaver by way of the `<noframes>` tag. You can put content in this tag so that browsers that don't support frames can still present alternative content to the user.

**NOTE**

If you do end up writing alternative content within a `<noframes>` tag, it's important that you limit the amount of CSS you decide to use. Chances are good that if a user's browser doesn't support frames, it probably will have limited support of CSS or won't support it at all. There are some exceptions—Older Pocket PCs, for instance, don't support frames but do support early CSS implementations.

---

To create content targeted to browsers that don't support frames, follow these steps:

1. With your IFrames page still open, place your cursor just after the closing `</iframes>` tag and choose Insert, HTML, Frames, NoFrames. The `<noframes></noframes>` tags are inserted.
2. Enter the text **This site uses Frames but your browser doesn't support them. Please come back using another browser!** between the tags. The code should resemble the following:

```
<noframes>
This site uses Frames but your browser doesn't support them.
Please come back using another browser!
</noframes>
```

3. When you've finished, save your page.

Now when a browser that doesn't support frames attempts to open this page, the text `This site uses Frames but your browser doesn't support them. Please come back using another browser!` is shown to the user instead of a blank page.

## Summary

Frames, framesets, and IFrames offer designers yet another option for laying out and structuring websites. When designing with these options, keep your target audience in mind. Will their experience at your site suffer because of the drawbacks framesets inherently expose? Will the features you provide enhance the user experience for ease of use and navigation? Ask yourself these questions when planning and moving forward with frame-based websites.

