

APPENDIX A

Accessibility

IN THIS APPENDIX

- ▶ Accessibility Standards
- ▶ Accessibility Reference

There is an implicit promise in the name of the World Wide Web—the promise of an information network that can be used by everyone around the world. The Web succeeds at what it does because of its universality. A web document is written in one or more web languages designed to be cross-platform and interoperable with a wide variety of technologies.

Clearly, the Web is designed to be open to a broad range of users, and that's where things start to get tricky! The average web user of moderate means in North America or Europe has a decently fast computer, a full-color monitor, a keyboard, a mouse, speakers, and a high-speed modem or faster connection. But that's just the average user—and people tend to be scattered all over the range of possibility, rarely conforming to the same capabilities. Welcome to designing on the Web!

In this appendix, I'll explain what web accessibility standards are, how they benefit both you and your users who have disabilities, and how you can use Dreamweaver to ensure that everyone can use your site.

Accessibility Standards

The way you design a website determines, to a very large extent, who is able to view the content on that site. If you're concerned only about those with the latest version of your favorite browser and the fastest hardware and

connection, there's no guarantee that you'll make a website that can be used by anyone who falls outside of those parameters.

There's a very large group of users who tend to fall outside of nearly everyone's target audience when developers design for the Web—users who have disabilities.

Web users who have visual disabilities are often stymied by web pages that rely on images, color, or visual layout to convey the meaning of the site's content. Those with limited vision have difficulties with low-contrast colors or small fonts. Deaf or hard-of-hearing users won't hear the soundtracks of multimedia. Users with limited physical dexterity might not be able to drag and drop or do other activities requiring a mouse. Pages with complex text that lack illustrations and summaries are very difficult for users with cognitive disabilities.

The Web isn't always easy to use if you have special needs. Some users, such as those who are blind, have to rely on special assistive technologies such as screen readers, Braille displays, or screen magnifiers for web access. However, these tools will work with your site only if you've carefully built your sites to allow access.

TIP

To learn more about how people with disabilities access the Web, visit the site of the International Center for Disability Resources on the Internet at www.icdri.org.

The process of creating a site that can be used by anyone—regardless of disability—is called *accessibility*. To properly create a website that is accessible, you'll need to know all about assistive technology, how people with disabilities use the Web, and how HTML and other web languages function in browsers. You'll also need to be an expert on accessibility's close cousin, *usability*, which is the study of how people use computers effectively.

Sound like a lot of work? Well, it is, believe me—but fortunately you won't have to do all that work yourself. The knowledge you'll need to construct accessible websites has been codified into accessibility standards, which function as a checklist of sorts so that you simply have to follow these rules to produce a site with no barriers to access.

Dreamweaver makes it even easier for you to follow those standards because they're built right into the software. By using Dreamweaver's accessibility features to create and check your work, you can greatly simplify the process of creating accessible websites.

Standards Resources

When it comes to the World Wide Web, there is one primary source for nearly all the standards you'll use—the World Wide Web Consortium (W3C). The W3C is an international association of some of the major players in the Web, from browser makers to research organizations. The official specifications for HTML, XML, XHTML, CSS, and other key web technologies were created by the W3C's working groups and released as recommendations for adoption on the Web.

TIP

The W3C's website is located at www.w3.org and is the definitive source for web specifications. However, most web specifications are incredibly dry reading, and unless you're some sort of masochist, you won't want to dive right into them. A better idea is to start at the website of the Web Standards Project at www.webstandards.org, a group of expert web developers who promote standards compliance.

One branch of the W3C concerns itself exclusively with access by people with disabilities—the Web Accessibility Initiative (WAI). Just as the W3C has produced standards for the HTML language, so has the WAI produced standards for accessibility.

Web Content Accessibility Guidelines

For web developers, the most important WAI standards are contained in the Web Content Accessibility Guidelines, or WCAG, which is a set of guidelines, checkpoints, and associated techniques that describe how to ensure the accessibility of your website.

TIP

You can read the full WCAG recommendation and download a checklist for easy reference from the W3C's website at www.w3.org/tr/wcag.

The WCAG recommendation lists 14 basic principles or guidelines that promote accessibility:

1. Provide equivalent alternatives to auditory and visual content.
2. Don't rely on color alone.
3. Use markup and style sheets and do so properly.
4. Clarify natural language usage.
5. Create tables that transform gracefully.
6. Ensure that pages featuring new technologies transform gracefully.
7. Ensure user control of time-sensitive content changes.
8. Ensure direct accessibility of embedded user interfaces.
9. Design for device independence.
10. Use interim solutions.
11. Use W3C technologies and guidelines.
12. Provide context and orientation information.
13. Provide clear navigation mechanisms.
14. Ensure that documents are clear and simple.

Each of these guidelines is supported by one or more checkpoints. For example, the checkpoints for guideline 2, "Don't rely on color alone," are

- ▶ **2.1**—Ensure that all information conveyed with color is also available without color, for example from context or markup.
- ▶ **2.2**—Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.

Each checkpoint is given a priority value. A priority 1 means that the failure to follow that checkpoint will exclude members of your audience with specific disabilities. Priority 2 checkpoints are designed to reduce the difficulty of access by people with disabilities, and priority 3 checkpoints actively improve the quality of access for individuals with special needs.

In WAI terminology, if your site fulfills all the priority 1 checkpoints, it is said to be Single-A compliant with WCAG. Meeting all priority 1 and 2 checkpoints grants your site Double-A status, and successfully meeting all the checkpoints qualifies a site as Triple-A level.

WCAG compliance levels have been accepted by many public and private organizations as the minimum requirement for sites they control. For example, California community college websites must meet at least WCAG Double-A standards.

Section 508

In addition to being directly adopted, the WCAG standard has been used to create specialized web accessibility policies. The most influential of these is the standard employed by the United States for most government websites.

The requirements for federal sites are described in Section 508, subsection 1194, of the 1998 amendments to the Rehabilitation Act. That's a mouthful to say at once, so everyone refers to the set of requirements simply as Section 508.

The aim of Section 508 is to ensure that government information technology is accessible to people with disabilities—both those working within federal agencies and those citizens who are using public web resources.

The Section 508 requirements for websites are modeled after the priority 1 checkpoints in WCAG, with a few modifications. Specifically, Section 508 adds some new requirements and eliminates a few priority checkpoints while generally rewriting from the technical recommendation language of the W3C to the form of bureaucratic regulation favored in government work.

TIP

The official website for Section 508 is located at www.section508.gov.

Which Standard to Follow?

It's been said that the great thing about standards is that there are so many to choose from. Despite the humor of this statement, there's still some truth to it—there's not one universal standard for accessibility but several, including Single-A WCAG, Double-A WCAG, Triple-A WCAG, and Section 508.

The overlap between Single-A WCAG checkpoints and Section 508 requirements remain very strong, however, so the techniques used to make a site accessible by one standard will generally ensure that the other standard is met.

The Double-A and Triple-A WCAG standards are harder to meet because they go beyond basic accessibility and require that web pages not be difficult to use.

In some cases, you might be able to choose which standard to follow. Most commercial and personal websites are unregulated, and thus you can select your level of compliance. Many commercial sites will aim for Single-A compliance, but Double-A compliance improves site access for disabled users or employees. Private organizations or corporations that provide services to people with disabilities will want to achieve Triple-A compliance.

As mentioned previously, public sector websites might have legal requirements for accessibility, depending on the location and type of public entity. For example, U.S. federal agencies such as the Department of Forestry are required to meet the Section 508 requirements, and universities in Australia, for instance, must meet WCAG Double-A. Your organization's legal or disability officer can advise you on specific regulatory obligations that apply to your website.

Conform with Standards

Conforming to accessibility standards provides many benefits. Besides reducing your potential legal complications (especially if you are subject to specific requirements), it can also improve the overall usability of your site because the considerations needed for producing an accessible website also lead to a site that is improved for everyone. For example, a transcript of an audio speech can benefit anyone accessing the web from a quiet public library.

Accessible standards also encourage designs that can be used on a diversity of web-accessible devices, including ATMs, set-top boxes (such as your cable or satellite box), Internet appliances (Xbox 360 or PlayStation 3), and personal digital assistants (PDAs such as the iPhone, SmartPhone, and BlackBerry). The same techniques that guarantee access for nonvisual browsers also improve access for users of text-only cell phones.

Creating an accessible website consists of ensuring that you've coded your site so that a broad audience can use it. Your audience includes not only traditional browsers and web devices, but also specialized programs or hardware collectively called *assistive technology*. Examples of assistive technology include screen readers, pointing devices, voice recognition software, screen magnifiers, Braille terminals, and onscreen keyboards.

Assistive technologies are usually innovative and clever approaches to overcoming obstacles, but like any computer feature, they can work only with what they're given in terms of information. If a Braille terminal encounters an image that isn't labeled properly (with an alt attribute), it cannot tell automatically if the image is a spacer GIF, a simple decoration, an important piece of content necessary for understanding the page, or a banner ad. As the author of a web page, you can provide this necessary information so that assistive technologies can function properly.

For example, the modified version of the CompanyEvents web page shown in Listing A.1, although a straightforward design, nonetheless has serious accessibility problems for users with disabilities.

LISTING A.1 The Inaccessible CompanyEvents Web Page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    margin: 0px;
}
.style1 {
    color: #993300
}
</style>
<head>
<title>Welcome to the Vecta Corporation</title>
<link href="styles.css" rel="stylesheet" type="text/css" />
</head>

<body bgcolor="#FFFFFF">
<div align="center">

<br />

<table width="467" border="0" cellpadding="0" cellspacing="0">
<tr>
<td>
    <div align="left"></div><br />
</td>
<tr>
<td valign="top">
<p align="left">
    Friday night was VectaCorp's annual "Concert at the
    Park". For those employees who missed the headlining
    band Spitalfield, below is a sample of what you missed.
</p>
<p align="left">
    <a href="Media/spitalfield.mp3" class="style1">
    Listen to audio clip (531 KB)
```

```
</a>
</p>
</td>
</tr>
</table>
<br /><br />
<p>
    Brought to you by the distinguished Marketing department<br />
    <a href="mailto:info@vectacorp.com" class="style1">
    Question or comments?
</a>
</p>
</div>
</body>
</html>
```

So what does the page look like in a browser? It's fairly simple (see Figure A.1) and displays perfectly well in a full-featured browser such as Internet Explorer.

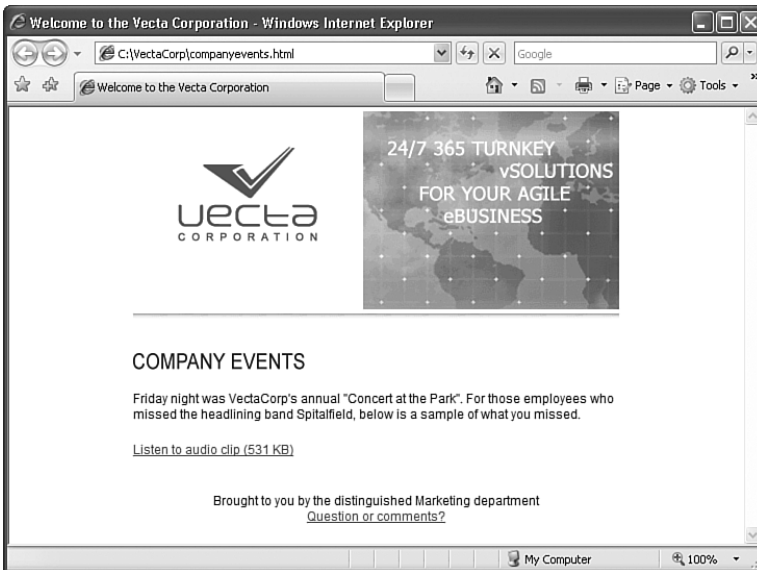


FIGURE A.1 The modified CompanyEvents web page looks fine in a full-featured browser such as Internet Explorer.

Note, however, that the red color used on the link to the audio clip doesn't reproduce well in the black-and-white screenshot printed in the book. What would this site be like for blind users? To test, we'll use a text browser named Lynx and view the page. Lynx displays all web pages without images or colors, just as plain text. This is a useful approximation of

what blind users experience when accessing a web page. Most users who can't see will use a screen reader program that reads out loud the text from a browser (such as Internet Explorer, Safari, or even Lynx) or a Braille display with raised dots. Both of these methods are roughly equal to the text display of Lynx. To install and view this page using Lynx, follow these steps:

1. Visit www.dreamweaverunleashed.com. Click the Resources link; then click the Software link from the navigation on the left. Next, click the link for Lynx Text Browser. Immediately download the Zip archive to your computer. The Mac version can be found at http://www.apple.com/downloads/macosx/unix_open_source/lynxtextwebbrowser.html.
2. When you've downloaded the Zip file, extract the Lynx 2.8.3 folder from the Zip archive.
3. Double-click `lynx.exe` to start the Lynx browser.
4. When the commands message appears, press the G key on your keyboard.
5. Type the location to your `companyevents.html` page: `C:\Vecta Corp\companyevents.html`.
6. Scroll to the top of the browser to see the text-only version of the `companyevents.html` page. As you can see from Figure A.2, some minor problems are clearly visible. Initially, the banner at the top isn't identified beyond `[header]` and the subheading for company events simply displays the text `[subheader_companyevents]`.

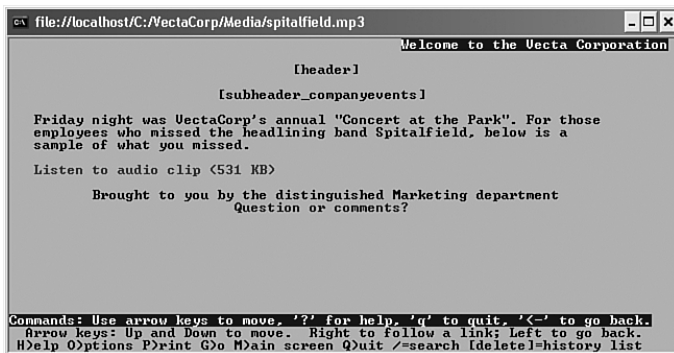


FIGURE A.2 The page is inaccessible when viewed in Lynx, a text browser.

To install the Lynx browser on a Mac, follow the steps outlined here:

1. Visit http://www.apple.com/downloads/macosx/unix_open_source/lynxtextwebbrowser.html and click the Download button, either from the top or bottom of the page, to begin downloading the Zip file to your computer.
2. When you've downloaded the Zip file, it will automatically extract, and you'll see the Lynx 2.8.6u disk image (DMG file) on your desktop (or in your Download stack in

Leopard). Double-click the DMG file to mount the disk image and then double-click the mounted image to see the contents.

3. Double-click Install to begin installation of the Lynx browser and type your Administrative password into Terminal when prompted.
4. You now have two options with the installation. You can leave it as is and run Terminal from /Applications/Utilities/Terminal.app, followed by typing Lynx and pressing Return, or you can copy Lynx.command into your Applications folder and double-click it to automatically open Terminal and launch Lynx for you. Whatever you're most comfortable with is what matters.
5. When the commands message appears, press the G key on your keyboard.
6. Type the location to your `companyevents.html` page: **/Vecta Corp/companyevents.html**, or wherever you might have copied the files, and hit Return.
7. Scroll to the top of the browser to see the text-only version of the `companyevents.html` page. Again, as you can see from Figure A.2, some minor problems are clearly visible. Initially, the banner at the top isn't identified beyond [header], and the subheading for company events simply displays the text [subheader_companyevents].

A further problem exists but might not be immediately obvious—the audio link presents a problem to deaf users. Because the audio file is available only in a MP3 file, anyone who is unable to hear sounds won't be able to hear the music.

In Listing A.2, you can see a revised version of the page. This version was created using the built-in accessibility check function in Dreamweaver, which is covered later in this appendix. The changes to the source code are shown in bold.

LISTING A.2 A More Accessible Version of the CompanyEvents Web Page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    margin: 0px;
}
.style1 {
    color: #993300
}
</style>
<head>
<title>Welcome to the Vecta Corporation</title>
<link href="styles.css" rel="stylesheet" type="text/css" />
```

```

</head>

<body bgcolor="#FFFFFF">
<div align="center">

<br />

<table width="467" border="0" cellpadding="0" cellspacing="0"
summary="Friday night was Vecta Corp's annual Concert at the
Park. If you missed the band Spitalfield play, read the
song lyrics for songs they sang live!">
<tr>
<td>
<div align="left"></div><br />
</td>
</tr>
<tr>
<td valign="top">
<p align="left">
Friday night was VectaCorp's annual &quot;Concert at the
Park&quot;. For those employees who missed the headlining
band Spitalfield, below is a sample of what you missed.
</p>
<p align="left">
<a href="Media/spitalfield.mp3" class="style1">
Listen to audio clip (531 KB)
</a><br />
<a href="spitalfield.html" class="style1">
Read the song lyrics</a>
</p>
</td>
</tr>
</table>
<br /><br />
<p>
Brought to you by the distinguished Marketing department<br />
<a href="mailto:info@vectacorp.com" class="style1">
Question or comments?
</a>
</p>
</div>
</body>
</html>

```

You'll notice that several new attributes such as `alt`, `longdesc`, and `summary` have been added. Furthermore, a transcript link was placed after the audio file link.

The revised page is shown in Lynx in Figure A.3. Although the changes aren't dramatic, they are enough to allow a broader group of users to access the page.

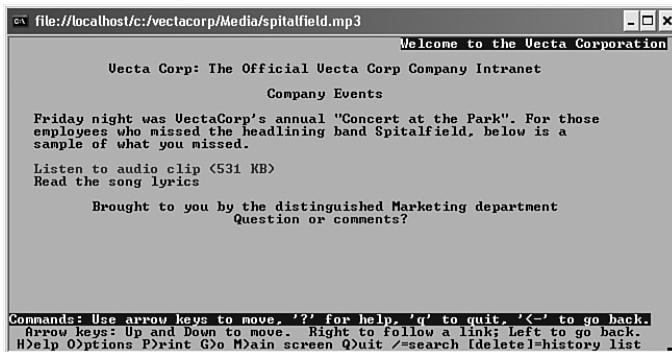


FIGURE A.3 Lynx can now view the page without errors, as can many users with varying disabilities.

Apply Standards to Sites

It's always easiest to make a web page or website accessible from the start and not have to spend time going back and redoing it from scratch. The effort of retrofitting is much harder than doing it right the first time. However, you might be dealing with older sites that need to be updated, or even sites that you didn't design and have inherited responsibility for. Dreamweaver assists you in bringing new and existing pages up to compliance with accessibility standards through accessibility dialog boxes and reports that analyze your page looking for specific problems. You can even run reports on all the pages in one folder on your hard drive or on the entire website. Let's explore these features now.

Accessibility Dialog Boxes

In earlier chapters you probably noticed those dialog boxes that kept appearing every time we added an image, a media element, or a form object. Those dialog boxes that we temporarily disabled early on were Dreamweaver's accessibility features hard at work. Dreamweaver uses these dialog boxes to prompt you automatically for required accessibility information. If you prefer to turn these dialog boxes on to be reminded of accessibility

features as you work, simply open the Preferences dialog box and set the necessary options by following these steps:

1. Open the Preferences dialog box by choosing Edit, Preferences (Dreamweaver, Preferences on the Mac).
2. Choose the Accessibility category to see the various accessibility options, shown in Figure A.4.

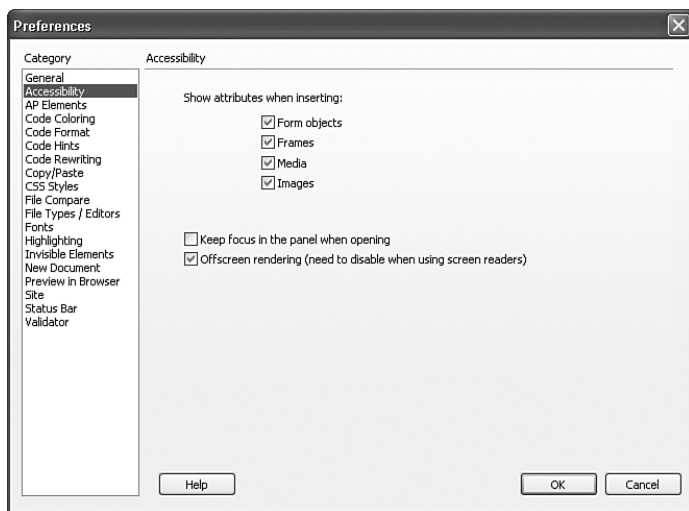


FIGURE A.4 Setting the accessibility options turns on/off accessibility dialog boxes.

3. Each of the four options—Form Objects, Frames, Media, and Images—turns on a different dialog box. When you insert one of those elements into your page, the dialog box appears and prompts you for information. For example, if you try to add an image, you'll see the dialog box pictured in Figure A.5.

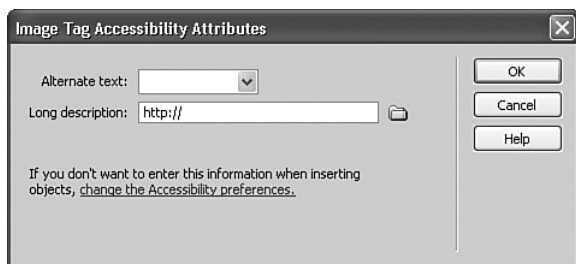


FIGURE A.5 The accessibility prompt for images requests alt and longdesc attributes.

4. Each dialog box requests a different set of accessibility-related attributes or information. These are shown in Table A.1.

TABLE A.1 Accessibility Dialogs Are Activated by Setting Accessibility Options

| Option (HTML Tag) | Accessibility Attributes |
|----------------------------|---|
| Image () | Alternative text (alt), Long Description (longdesc) |
| Frame (<frame>) | Frame Title (title) |
| Form (<input>, <textarea>) | Label (<label>), Style (nesting of <label>), Position (location of <label>), Access Key (accesskey), Tab Index (tabindex) |
| Media (<object>) | Title (title), Access Key (accesskey), Tab Index (tabindex) |
| Table (<table>) | Caption (<caption>), Align Caption (<caption align>), Summary (summary), Header (scope) |

You’re probably already familiar with the alt attribute; this is a text replacement for the image. An alt attribute isn’t a description of the image but a functional replacement for it. If the image has no function beyond decoration, the alt value should be alt=" ". (If the alt attribute is left blank, the name of the image is repeated to the user.) For little bullet icons, use alt="*", not alt="red circle". And definitely don’t use the name of the graphic, such as alt="redbullet.jpg". (Users with disabilities don’t care what the name of the image is; they want the description of it.)

The longdesc attribute is used to provide a description of an image; unlike alt, longdesc is not a text value, but the URL of a page that describes the image in text. A longdesc should be used if the image contains information that isn’t shown by the alt text, such as a chart or a graph. It can also be used to describe the contents of photographs or paintings.

The title attribute is a name or short description of a frame or object that is meant to be read to a human. A frame usually has a name attribute, but this is used by the browser to identify the frame and isn’t necessarily written to make sense to the user. For example, name="mnav" is confusing. The title should be clear and understandable and describe the function of the frame or object, such as name="Main Navigation Panel".

The accesskey and tabindex attributes are used to enable improved keyboard navigation. The accesskey attribute designates a specific key that can be pressed in conjunction with the modifier key—usually the Control or Alt key—to activate a link or object. The tabindex key sets an order for tabbing through links and objects; pressing the Tab key advances you through the page in order of the tabindex attributes.

The <label> attribute<attribute><\$I<label>attribute (accessibility);description of > tag provides a text label for form controls, such as text fields or check boxes. You can

determine the position of the label tags using the label settings on the dialog box. The `<label>` is important for screen-reader users who need to know what each form field does when they can't rely on visual layout clues.

5. Tables have a number of attributes, such as `scope` and `summary`. In general, the `summary` attribute is used to describe the contents of the table as a whole, whereas the `scope` attribute is used to describe the relationship between the table header and the contents of the table cells. In addition, the `<caption>` tag can be used to add a caption to the table.

TIP

For more information on HTML tags and attributes used to make pages more accessible, be sure to visit the excellent accessibility tutorials at the Web Accessibility In Mind (WebAIM) site at www.webaim.org.

6. Finally, as you'll see in Appendix C, "Defining Preferences," Windows users can use the last two check boxes in the Preferences dialog box to maintain focus on the Accessibility panel when a specific object such as a form object, frame, media element, or image is inserted. Furthermore, you can disable the Offscreen Rendering check box if your page isn't displaying properly within a screen reader.

Check Accessibility

The accessibility report built into Dreamweaver is set to check against both WCAG and Section 508 standards. The WCAG standard is checked against Single-A and Double-A level accessibility; Triple-A checkpoints aren't tested.

To check the accessibility of a page you're working on, first save the page. Then select File, Check Page, Accessibility. Selecting this option will generate an accessibility report on your existing page.

NOTE

The Check Accessibility command can be used only after you've saved the file you're working on. If you don't save before checking, the accessibility report won't reflect any recent changes.

An example of the output of an accessibility report can be seen in Figure A.6. This report was run on the web page in Listing A.2, the corrected CompanyEvents web page.

The accessibility check function runs an analysis of each part of your web page, testing it against certain criteria called *accessibility checks*. For each one, it gives one of three results: pass, fail, or can't determine. If your page fails a check, you'll need to correct that to improve the accessibility of the page. A failed test is represented by a red X in the accessibility report.

| File | Line | Description |
|----------------------|------|---|
| ? companyevents.html | 3 | Color is not essential [508 c/WCAG 2.1 P1] -- MANUAL -- |
| ? companyevents.html | 3 | Colors are visible [508 c/WCAG 2.2 P1] -- MANUAL -- |
| ? companyevents.html | 16 | Style sheets should not be necessary [508 d/WCAG 6.1 P1] -- MANUAL -- The page uses style sheets to present its content. ... |
| ? companyevents.html | 4 | Style sheets should not be necessary [508 d/WCAG 6.1 P1] -- MANUAL -- The page uses style sheets to present its content. ... |
| ? companyevents.html | 26 | Data table should have headers [508 g/WCAG 5.1 P1] -- MANUAL -- The page contains a table that might be used to present... |
| ? companyevents.html | 19 | Skip repetitive links [508 o/WCAG 13.6 P1] -- MANUAL -- Check if a textual link is present for skipping navigation links that ar... |
| ? companyevents.html | 19 | GIFs do not cause the screen to flicker [508 j/WCAG 7.1 P1] -- MANUAL -- The page contains GIF images: Are they causing t... |
| ? companyevents.html | 19 | Use clear language for site's content [WCAG 14.1 P1] -- MANUAL -- |
| ? companyevents.html | 19 | Clarify natural language usage [WCAG 4.1 P1] -- MANUAL -- |
| ? companyevents.html | 44 | Linked AUDIO with equivalent CONTENT [508 a/WCAG 1.1 P1] -- MANUAL -- "A" tag points to an audio file. |
| ? companyevents.html | 21 | Ensure sufficient contrast between foreground and background colors [WCAG 2.2 P2] -- MANUAL -- |
| ? companyevents.html | 32 | Ensure sufficient contrast between foreground and background colors [WCAG 2.2 P2] -- MANUAL -- |
| ? companyevents.html | 3 | Ensure sufficient contrast between foreground and background colors [WCAG 2.2 P2] -- MANUAL -- |
| ? companyevents.html | 21 | Use markup rather than images [WCAG 3.1 P2] -- MANUAL -- |
| ? companyevents.html | 32 | Use markup rather than images [WCAG 3.1 P2] -- MANUAL -- |
| ? companyevents.html | 3 | Document should be valid with respect to published grammars [WCAG 3.2 P2] -- MANUAL -- Use the Markup Validator to check... |
| ? companyevents.html | 3 | Document should be valid with respect to published grammars [WCAG 3.2 P2] -- MANUAL -- Use a CSS validator to check the C... |

FIGURE A.6 The accessibility report identifies problems and potential problems.

CAUTION

An automated checking program can do only so much; there's no perfect way to make software fix web pages. It's possible for a page to pass every automated test and still be inaccessible. For this reason, read up on accessibility techniques at the W3C's Web Accessibility Initiative site (www.w3.org/WAI/) and consider acquiring for your own testing purposes one of the programs used by people with disabilities.

Even with the accessibility features that we did add to the page, you can see that the checking program still identified potential problems. To make the page fully accessible, it would be wise to review each feature and correct it accordingly.

Manual Checks

If the accessibility report has a question mark for the result of a test, it usually means that human judgment is needed to determine whether a test was passed. This is known as a *manual check*.

A good example of a manual check is the `alt` attribute for an image. The computer can tell if the `` tag has an `alt` attribute, but it isn't able to determine whether the `alt` attribute is accurate. The purpose of the question mark is to tell you to evaluate the question yourself to determine whether accessibility issues occur on the page based on a judgment call you make.

Sitewide Accessibility Reports

To test a large number of web pages, you don't have to individually load each one and run an accessibility report. Instead, you can use the site report function. This lets you select whether to run an accessibility report on the current page, the entire website you're working with, selected files in that site, or all the web files in a folder.

To use the site reports, choose Site, Reports. You'll see the choices shown in Figure A.7. Be sure to check the box for Accessibility and choose the appropriate files to test from the pull-down menu.

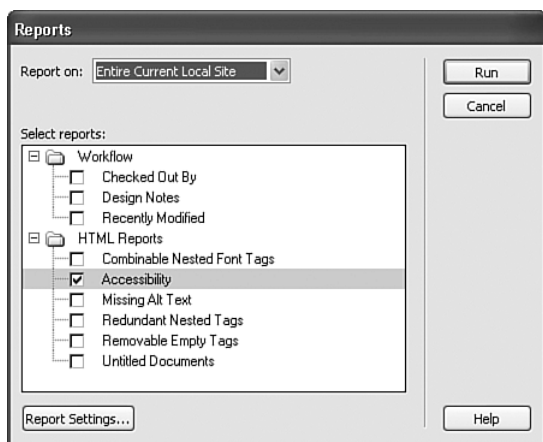


FIGURE A.7 You can test an entire site or a folder of HTML pages at once using the site reports function.

As you saw in Chapter 11, “Enhancing Workflow,” in addition to specifying the files to be checked, you can also set report parameters to include or exclude certain accessibility checks. By default, the accessibility report checks both the WCAG Single-A (also known as Priority 1) and Double-A (also known as Priority 2) standards and the Section 508 standard. To change this, highlight the accessibility report option by clicking the word *Accessibility* and then click the Report Settings button.

This action calls up the Accessibility options shown in Figure A.8. You can toggle open the list of options by clicking the (+) icons beside each category. Using the Enable or Disable buttons, you can customize your report to check only the tests, or groups of tests, that matter to you. You can also set the report to list all checks performed, not just those that were failed or that need human judgment.

TIP

If you need more advanced accessibility evaluation and repair features, you might want to look at automation software provided by a group of third-party organizations such as UsableNet. For a list of these companies and the automated accessibility software they offer, visit www.adobe.com/macromedia/accessibility/usablenet.html.

Accessibility Reference

The rest of this appendix is a reference to the checks performed by the accessibility checker in Dreamweaver. Each test is identified by a short title, but it’s not always clear what each title means; the list that follows will clarify the meaning of the test titles.

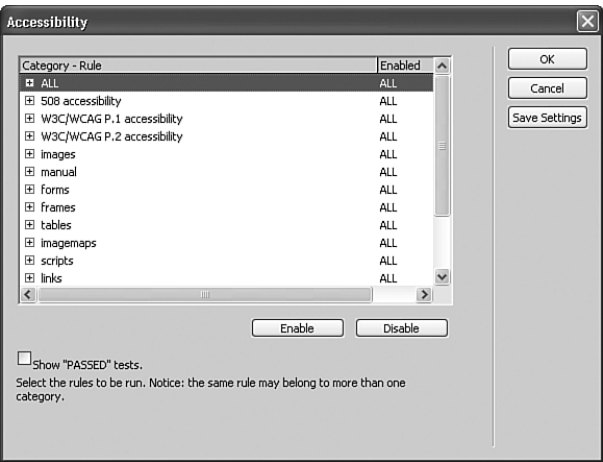


FIGURE A.8 You can turn off or on specific accessibility tests in the report options.

NOTE

Dreamweaver comes with a built-in reference that is useful for understanding accessibility guidelines. To access this reference, choose Window, Result, Reference and load the UsableNet Accessibility reference.

Image Tests

Images, because they're visual, can present serious obstacles to users who can't see. There is also some danger that a strobing image could trigger seizures in photo-epileptic users. The tests that are run on images are shown in Table A.2.

TABLE A.2 Accessibility Checks Performed on Images

| WCAG | Section 508 | Accessibility Test |
|------|-------------|---|
| 1.1 | 1194.22(a) | Spacer IMG with valid ALT |
| 1.1 | 1194.22(a) | No LONGDESC for spacer IMG |
| 1.1 | 1194.22(a) | Non spacer IMG with valid ALT |
| 1.1 | 1194.22(a) | Non spacer IMG with equivalent ALT |
| 1.1 | 1194.22(a) | Non spacer IMG with valid LONGDESC |
| 1.1 | 1194.22(a) | Non spacer IMG needs LONGDESC |
| 1.1 | 1194.22(a) | Image OBJECT with valid CONTENT |
| 1.1 | 1194.22(a) | Image OBJECT with equivalent CONTENT |
| 7.1 | 1194.22(j) | GIFs do not cause the screen to flicker |

A *spacer image* is one that serves only to lay out the page and doesn't contain any useful information itself. Most of these are blank or transparent images. Any purely decorative image, such as a spacer image, should have an alt attribute value of alt="".

In the preceding table, some of these tests seem to be repeated with just a subtle change; for example, Non spacer IMG with valid ALT and Non spacer IMG with equivalent ALT. A *valid* alt attribute is simply one that exists. If you leave off the alt attribute and give no value at all, it's not valid.

However, a *valid* alt attribute is not necessarily an *equivalent* alt attribute. Consider the top banner, which contained the text *Welcome to the Vecta Corporation*. If the alt value was alt="Welcome", this would be a valid alt attribute, but it would not be an equivalent value. An equivalent value in this case would be alt="Welcome to the Vecta Corporation".

An automatic program, such as the accessibility checker in Dreamweaver, can check to see whether an alt attribute is valid—but only human judgment can determine whether the value is equivalent. For this reason, there is a manual check that goes with some automatic checks. The longdesc attribute is another example—only a human can determine whether additional information is needed to convey the image content.

CAUTION

You might begin to think that images are the enemy of accessibility and should be avoided. Nothing could be further from the truth! Images, when given appropriate alt and longdesc attributes, are not an accessibility problem. In fact, lack of images can introduce accessibility hurdles for some people, including those with problems reading because of cognitive disabilities. A good illustration really is worth a thousand words, so don't be afraid to use images!

Imagemap Tests

Imagemaps share all the possible pitfalls that could accompany images and introduce several potential problems of their own. The special checks done on imagemaps are shown in Table A.3.

TABLE A.3 Accessibility Checks Performed on Imagemaps

| WCAG | Section 508 | Accessibility Test |
|------|-------------|---|
| 1.1 | 1194.22(a) | AREA with valid ALT |
| 1.1 | 1194.22(a) | AREA with equivalent ALT |
| 1.2 | 1194.22(e) | Links are needed for server-side imagemap |
| 9.1 | 1194.22(e) | No server-side image maps should be used |

There are two types of imagemaps in HTML—client-side imagemaps that use <area> tags to define shapes, and server-side imagemaps that require scripts to determine the outcome of a map click.

Of the two, client-side maps are much more accessible because assistive technology programs can read the `<area>` tags and create a menu instead of an image with hotspots. However, each `<area>` must be marked with an appropriate `alt` attribute.

Server-side imagemaps present serious accessibility problems for users who can't see images and, thus, should be avoided whenever possible. If you do use a server-side imagemap, you should make sure to provide equivalent text links for every hotspot on the imagemap.

Color and Style Tests

As shown in the `companyevents.html` example, the use of color can create accessibility problems when used carelessly. Contrast is important as well; blue links on light blue backgrounds are hard to see. Style sheets are almost always visual and might have many of the same problems as color when used to convey specific information. The checks for color and style sheets are shown on Table A.4.

TABLE A.4 Accessibility Checks Related to Colors and Style Sheets

| WCAG | Section 508 | Accessibility Test |
|------|-------------|---------------------------------------|
| 2.1 | 1194.22(c) | Color is not essential. |
| 2.2 | 1194.22(c) | Colors are visible. |
| 6.1 | 1194.22(d) | Style sheets should not be necessary. |

Please keep in mind that these checks are not saying “don't use color” or “don't use CSS.” In fact, you most assuredly should use both of them—and use them regularly. Color provides many usability and comprehension benefits, and style sheets are a boon to accessibility. These tests merely ask you to ensure that the vital information of the page isn't conveyed *only* by a style or color choice and is shown on the page in some other manner.

Form and Scripting Tests

Forms and scripts can present problems to assistive technology programs such as screen readers. The checks done for forms and scripts are shown in Table A.5.

TABLE A.5 Accessibility Checks for Forms and Scripts

| WCAG | Section 508 | Accessibility Test |
|------|-------------|----------------------------------|
| 1.1 | 1194.22(a) | INPUT with valid ALT. |
| 1.1 | 1194.22(a) | INPUT with equivalent ALT. |
| 1.1 | 1194.22(a) | SCRIPT with valid NOSCRIPT. |
| 1.1 | 1194.22(a) | SCRIPT with equivalent NOSCRIPT. |
| | 1194.22(l) | Scripts are accessible. |
| 6.5 | | No JavaScript links are used. |
| 7.4 | 1194.22(p) | No auto refresh is used. |

The requirement for `<input>` tags to have `alt` attributes applies only to image Submit buttons—those `<input>` tags with `type="image"`.

Scripts that have an effect, such as presenting new content, should have an equivalent `<noscript>` tag that either provides access to the content or links to a page or server-side program that has the same effect. Scripts that validate input or produce cosmetic effects such as mouse-overs aren't required to have `<noscript>` tags.

Links that are purely JavaScript actions—or pull-down menus that change the current location without a Submit button being pressed—can be very difficult for assistive technologies and should be avoided. Also, pages that automatically refresh based on `<meta>` tags can disrupt screen readers; instead, use HTTP redirects in the server configuration or `.htaccess` file.

Table and Frame Tests

Tables and frames are visual ways of presenting content in specific locations. When used injudiciously, they can introduce serious accessibility errors for people with visual disabilities who might not be able to see the page at all or who might be using a screen magnifier and can't see the entire layout at once. The tests for tables and frames are listed in Table A.6.

TABLE A.6 Accessibility Checks Performed on Tables and Frames

| WCAG | Section 508 | Accessibility Test |
|------|-------------|---|
| 5.1 | 1194.22(g) | Data table should have headers. |
| 5.1 | 1194.22(g) | Cell of data table should refer to headers. |
| 5.1 | 1194.22(g) | Data tables should be defined by TABLE tag. |
| 5.1 | 1194.22(g) | Multiple headers should be marked in data tables. |
| 12.1 | 1194.22(i) | FRAME with valid TITLE. |
| 12.1 | 1194.22(i) | IFRAME with valid TITLE. |

The tests listed for tables apply to data tables—tables that have been inserted to display tabular columns of information, such as a bus schedule. Web accessibility standards distinguish between data tables and layout tables, which are tables used to lay out web pages in two dimensions on the screen. Only data tables require special coding for headers—and then only when the table is complex.

As mentioned earlier, a title attribute is meant to be a human-understandable name, such as `title="Navigation Frame"` or `title="Banner Ad Frame"`. Avoid naming your tables by their location; `alt="Left Frame"` is useless because it doesn't describe the function, just the location.

CAUTION

Should you even use tables and frames for layout? Tables used to be a more serious accessibility problem when screen readers would read across line by line, cutting cells in strange places. Current screen readers have improved this, and all you have to do is make sure that your table cells make sense when read in the order they appear in the source code.

Frames are more problematic. As you'll read in Appendix D, "Working with Frames and Framesets," apart from potential accessibility hurdles, frames can introduce problems with book marking and usability. However, if labeled correctly, and if an appropriate `<noframes>` tag is provided, frames can be made accessible as well.

This doesn't mean that they're the best solution—often, a nonframed design with CSS for layout can accomplish as much as tables or frames and has even greater accessibility. Use tables and frames with care, if you decide to use them at all.

Multimedia and Applet Tests

Multimedia, as used here, refers both to video and audio; embedded objects can include Java applets, Flash animations, and more. The tests for these types of content are shown in Table A.7.

TABLE A.7 Accessibility Checks for Multimedia Files and Applets

| WCAG | Section 508 | Accessibility Test |
|------|-------------|---|
| 1.1 | 1194.22(a) | Audio/video OBJECT with valid CONTENT. |
| 1.1 | 1194.22(a) | Audio/video OBJECT with equivalent CONTENT. |
| 1.1 | 1194.22(a) | OBJECT with valid CONTENT. |
| 1.1 | 1194.22(a) | OBJECT with equivalent CONTENT. |
| 1.4 | 1194.22(b) | Multimedia with synchronized alternative. |
| 1.3 | 1194.22(b) | Multimedia with equivalent audio description. |
| 1.1 | 1194.22(a) | Linked AUDIO with equivalent CONTENT. |
| | 1194.22(m) | Link to plug-in is present. |
| 1.1 | 1194.22(a) | APPLET with valid ALT. |
| 1.1 | 1194.22(a) | APPLET with valid CONTENT. |
| 1.1 | 1194.22(a) | APPLET with equivalent ALT. |

In general, the easiest way to deal with multimedia is to provide a text transcript of the information. In addition to the dialog box, action and events must be described. A *synchronized alternative* is a text or audio version that plays at the same time as the video,

such as a caption or an audio description. The synchronization is usually accomplished by using the Synchronized Multimedia Integration Language (SMIL).

TIP

To learn more about SMIL, visit the W3C’s multimedia page at www.w3.org/AudioVideo on the Web.

Other Accessibility Tests

Several other accessibility checks that are performed don’t fall into separate categories but, nevertheless, are very important for ensuring the accessibility of your site. These are shown in Table A.8.

TABLE A.8 Additional Accessibility Checks Performed by Dreamweaver

| WCAG | Section 508 | Accessibility Test |
|------|-------------|--|
| 13.6 | 1194.22(o) | Skip repetitive links. |
| 7.1 | 1194.22(j) | Avoid causing the screen to flicker. |
| 14.1 | | Use clear language for site’s content. |
| 4.1 | | Clarify natural language usage. |
| | | Proprietary tags are used. |
| 6.2 | 1194.22(k) | Text-only equivalent page might be needed. |

Repetitive links are the pet peeves of many screen reader users. When a visual browser loads a web page, a sighted user can instantly scan it in a glance, jumping to the content—usually in the middle of the page—and ignoring the navigation bars. Screen reader users don’t have this luxury; they have to listen to all the links, on every page, again and again before reaching the content. For this reason, the web accessibility standards suggest a Skip Navigation link at the top of the page that takes the user directly to the main content, bypassing the navigation bars.

NOTE

For a great tutorial on creating skip navigation functionality, visit Jim Thatcher’s Skip Navigation tutorial at www.jimthatcher.com/skipnav.htm.

In accessibility standards terminology, a *natural language* is any language that a human being speaks or writes. When part of a page is written in a different language, this could confuse screen readers or automatic translation software; therefore, changes in natural language should be shown in the HTML tags. Use the lang attribute (and xml:lang in XHTML) to indicate changes in language, such as this:

```
<p>I counted to three:  
  <span lang="es">Uno, dos, tres.</span>  
</p>
```

If you've tried everything and you can't make a web page accessible, you can make an equivalent page that is simpler and presents the same information in a straightforward markup language. This is often called a *text-only page*, but in general, a text-only page isn't necessary. Nearly any page can be made accessible by adding a few extra tags and attributes.

Summary

By employing the techniques of accessible web design, you can ensure that your users with disabilities won't be shut out from accessing your website. These techniques are described in the web accessibility standards.

The World Wide Web Consortium's Web Content Accessibility Guidelines define the technical considerations for creating accessible websites. The WCAG checkpoints provide you with a blueprint for your accessible website and have been adopted (in modified form) by the United States government in the form of the Section 508 requirements.

Dreamweaver enables you to apply these accessibility standards to your new web designs or to existing websites. Accessible templates and prompts for accessibility attributes let you design for accessibility from the start, and integrated site reports can spot accessibility problems in one HTML document or on an entire site. A complete reference to accessibility rules is available at any time through Dreamweaver's Reference panel.

Creating an accessible website shouldn't be an extra chore—it should be part and parcel of your good web design practices. The special accessibility functions of Dreamweaver help make these important practices quick and easy to apply.

