# Using an Access-Centered Design to Improve Accessibility: A Primer for Technical Communicators

*Linda Enders Roberts*

*This paper discusses accessibility barriers as defined by the World Wide Web Consortium (W3C) and provides a primer on how technical communicators can remove these barriers during their Web design process and test to ensure the barriers were removed. The article focuses on 10 common barriers to a meaningful experience for people with disabilities, barriers that a technical communicator can consider when designing online information. Working on accessibility issues before online information goes live will help to reduce re-work and re-design and can save a lot of headaches for a technical communicator.*

## INTRODUCTION

This paper discusses accessibility barriers as defined by the World Wide Web Consortium (W3C) and provides a primer on how technical communicators can remove these barriers during their Web design process and test to ensure the barriers were removed.

The W3C launched its Web Accessibility Initiative (WAI) in April 1997. In May of 1999, the WAI presented as a formal recommendation the Web Content Accessibility Guidelines (WCAG) 1.0 to address the issue of accessibility and the Internet. This document presents 14 formal guidelines on making Web content more accessible to people with disabilities. Spread among these guidelines are 65 checkpoints, each of which has been assigned one of three priority levels. A Web page *must* satisfy priority 1 checkpoints to attain a minimum degree of accessibility, *should* satisfy priority 2 checkpoints for a higher degree of accessibility, and *may* address priority 3 checkpoints for maximum accessibility and usability (Slatin and Rush 2003, 43).

As Slatin and Rush (2003, 6) note, even if a content developer creates a Web site that conforms to all of WCAG 1.0's guidelines, that site may not be usable to a person with disabilities. Because of this potential problem, Slatin (2001, 76) proposes an "access-first design" philosophy in creating online information. He feels that this philosophy logically extends from a user-centered design philosophy. User-centered design has the users—real people—as the focus around which a product is developed instead of focusing on the product's functions and features. In an access-first design approach, a technical communicator designs online information that will provide a meaningful experience for people with disabilities, or one in which people with disabilities are able to access and use a site as effectively as people without disabilities (Slatin and Rush 2003, 10).

So what features should an online communicator focus on in access-first design? The World Wide Web Consortium (1999a 2001) spells out common barriers to a meaningful experience for people with disabilities, barriers that a technical communicator can consider when designing online information. These barriers include

1. Images without alternative text
2. Lack of alternative text for imagemap hot-spots
3. Uncaptioned audio or undescribed video
4. Lack of alternative information for users who cannot access frames or scripts
5. Lack of alternatives for applets and plugins
6. Lack of logical text for hyperlinks
7. Misleading use of structural elements on pages
8. Tables that are difficult to decipher when linearized
9. Sites with poor color contrast
10. Lack of testing

## BARRIER 1: IMAGES WITHOUT ALTERNATIVE TEXT

People with disabilities such as blindness or low vision rely on assistive technologies (AT) such as Freedom Scientific's JAWS screen reader, IBM's Home Page Reader, Dolphin Systems' HAL, or ALVA Access Group's brailleOUT to read the information on a Web page to them or to send the information to a Braille writer. WCAG 1.0 checkpoint 1.1 states that if a Web page contains "images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ASCII art, frames, scripts, images used as list bullets, spacers, graphical buttons … [Priority 1]" (World Wide Web Consortium 1999c), then a content designer needs to provide a concise and descriptive text equivalent to describe the function of each graphic, image, and animation. Content designers can comply with this guideline by adding an alt attribute to the <img> elements (for images, list bullets, and spacers) or the alt attribute to the <area> elements (for image maps).

This task seems like it should be easy to accomplish, but Slatin and Rush (2003, 248-249) caution content developers about the information they provide in an alt attribute. The authors suggest avoiding phrases such as "Link to next page," "Go to the next page," and "Click here to go to the next page" when providing text in the alt attribute for graphical links. Instead, they suggest using the text "Next page." They note that words such as "link to" are unnecessary because ATs such as JAWS say the word "link" before reading each link in a Web page and also because these extraneous words eat up valuable time. Also, ATs such as JAWS have a feature that allows users to open a dialog box and navigate the links on a Web page in alphabetical order. If the text in the alt attributes for all the links started with the words "Go to," then this feature would be rendered rather useless. Finally, contextual references such as "click here" have little meaning when they are read aloud.

If an image or animation requires a longer description (over a few sentences), a content designer can also include the longdesc attribute within an <img> element to provide a link to an external Web page or to a file that contains a longer description of the content and function of the graphic. This link information is invisible to people who are not using screen readers. Before the longdesc attribute was available, some content designers used a "d-link" (descriptive link) to provide a link to an external Web page or file that would contain a longer description of the graphic. With this technique, the letter "d" is placed close to an image and can be selected.

Some experts argue that descriptions for some images should not be provided. If a Web page's layout uses items like 1-pixel by 1-pixel spacer GIF files or transparent images, for example, the alt attributes for these layout items should be set to null (alt="") so that screen readers will skip over them. A person using an AT would never know that those layout items were on the Web page (Slatin and Rush 2003, 251-252). On the other hand, experts caution that content designers should not omit the alt attributes for any images thinking that these images will be skipped over by ATs if they do so. If an alt attribute in an <img> element is left out, then an AT may just say "picture" or "graphic—untitled," or read the contents of the src attribute, which is the name and physical location of the image file. Such information can leave a user scratching his or her head or be very long and confusing.

For decorative items on a Web page such as bullets that are GIF files, experts also suggest using a null alt attribute. Theofanos and Redish (2003, 41) write, "Blind users also object to listening to descriptions of elements, such as decorative bullets that add no meaning to the page and that make them wait through three words to get to the real meaning." Just as sighted users scan a Web page's content to quickly find the information they seek, people using a screen reader scan with their ears and do not want to hear a lot of extraneous information.

If a Web page uses ASCII graphics to display a chart or a graphic, WCAG 1.0 checkpoint 1.1 also notes that the Web developer needs to provide a text equivalent of the picture. The World Wide Web Consortium (2003) defines ASCII art as "text characters and symbols that are combined to create an image." Common examples of ASCII art include charts and graphs created in text-only outputs, emoticons used in e-mail to express feelings, and pictures created with text that can sometimes be found in the signature files of e-mails. The text equivalent for a simple graphic like a picture in a signature file could consist of a brief description or the title of the figure. For a complex chart, a text link to a Web page containing long description of the chart could be included.

## BARRIER 2: LACK OF ALTERNATIVE TEXT FOR IMAGEMAP HOT-SPOTS

The previous section discussed the alt attribute text for images, spacer GIFs, and ASCII art. WCAG 1.0 checkpoint 1.1 also addresses the need to include an alt attribute for each image map region. The World Wide Web Consortium (2003) defines an image map as "an image that has been divided into regions with associated actions. When a user clicks on an active region of a client-side image map, the user agent (such as a browser, mobile phone, some ATs, etc.) calculates in which region the click occurred and follows the link associated with that region."

Nielsen (2000) notes that content developers should add text to the alt attribute associated with each <area> element so that users who cannot see the image map will be presented with verbal descriptions of each destination as they move the cursor around the image map.

If the regions shown in an image map do not have text in their alt attributes, a screen reader like JAWS reads aloud the contents of the href attribute, the name and path associated with each link. Based solely on this usually cryptic file information, a person using a screen reader would probably have a hard time making a navigation choice.

 

## BARRIER 3: UNCAPTIONED AUDIO OR UNDESCRIBED VIDEO

Just as WCAG 1.0 checkpoint 1.1 applies to images (as discussed in the previous section), it also applies to audio and video files. This checkpoint states that a content designer should provide a text equivalent for "sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video" (World Wide Web Consortium 1999c). This part of checkpoint 1.1 can be satisfied by making a text transcript of the audio file or the audio portion of the video presentation available. Slatin and Rush (2003, 406) define text transcripts as a "text version of the audio portion—the dialog and other sounds—of the video presentation." This information is usually provided using a link to a separate Web page and can be accessed when the audio or video functionality is turned off or is not supported.

WCAG 1.0 checkpoints 1.3 and 1.4 also apply to this issue. Checkpoint 1.3 states that a content designer should "provide an auditory description of the important information of the visual track of a multimedia presentation. [Priority 1]" (World Wide Web Consortium 1999c). Checkpoint 1.4 states that "for any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation. [Priority 1]" (World Wide Web Consortium 1999c). Slatin and Rush (2003, 371) state that these two checkpoints can be satisfied by using synchronized close captioning or synchronized audio description.

Synchronized closed captioning is provided for people who cannot hear, have the sound on their computers muted, or have their computer speakers turned off. In closed captioning, a content designer provides a text track of the video's dialog and other sounds so that this text track is synchronized with the video's soundtrack. Ideally, this text track is a word-for word (or as near as possible as screen time permits) transcript of the video's sound track (Slatin and Rush 2003, 371-373). Many video editing programs allow a designer to provide a closed caption track. WGBH in Boston pioneered the use of closed captioning in the 1970s, and they make available a program named MAGpie (http://ncam.wgbh.org/webaccess/magpie/) that allows a content designer to add captions to a video.

Slatin and Rush (2003, 394) define a synchronized audio description as "a spoken description of the activities presented through video." This information is recorded on a separate track from the video and inserted in the natural pauses found in the video soundtrack. It provides a description of key visual elements such as the scene, characters, body language, and visual elements. This descriptive information helps users who cannot see the action understand the context of the video.

## BARRIER 4: LACK OF ALTERNATIVE INFORMATION FOR USERS WHO CANNOT ACCESS FRAMES OR SCRIPTS

WCAG 1.0 checkpoint 1.1 also notes that if a Web page uses frames or scripts, a content designer should provide a text alternate or alternate functionality for users who have frames or scripts disabled or who are using a browser that does not support frames or scripts.

For Web sites that use frames, a text alternate can be provided using the <noframes> element. In the <noframes> element, a Web designer can provide a link to an unframed version of the site for users using ATs or who are using user agents that do not support frames. A browser that supports frames simply ignores the information provided between the starting <noframes> element and the ending </noframes> element.

Nielsen (2000) urges content designers not to use the <noframes> element to simply provide a link to a Web site that allows a user to download a browser that does support frames. He feels that this approach is unhelpful because a user almost certainly does not want to download a large program simply to find out what is in a Web site. A true second version of the site without frames would be more helpful.

For a Web site that uses scripts, include the <noscript> element to provide equivalent functionality. Between the opening <noscript> and closing </noscript> elements, a content designer can include graphics, links, text, forms, etc. to provide an alternative to the script. Like the <noframes> element, the information contained in the <noscript> element is ignored if the user's browser supports scripts.

## BARRIER 5: LACK OF ALTERNATIVES FOR APPLETS AND PLUGINS

WCAG 1.0 checkpoint 1.1 also notes that if a Web page is constructed using applets and programmatic objects, a content designer should provide a text alternate for users whose browsers do not support the <applet> or <object> elements. This text lets users know that these objects are on a Web page and provides some information about the content of the objects. The World Wide Web

Consortium (1999b) notes that while applets may be included in a document using either the <applet> or <object> element, the <object> element is preferred.

For a Web site that uses the <applet> element, a text alternative can be provided using the alt or longdesc attribute within the <applet> element. For Web sites that use the <object> element, a text alternative can be provided within the body of the <object> element.

WCAG 1.0 checkpoint 6.3 states, "Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page. [Priority 1]" (World Wide Web Consortium 1999c). Slatin and Rush (2003, 424-426) note that the <object> element, unlike the alt attribute, can contain other HTML code. So, content developers are able to provide links to additional information, graphics, and text from within the <object> element. This information is only seen by users whose browsers do not support the display of multimedia objects or users who have disabled this feature within their browsers. For example, if a movie or animation depicts a flow chart of a process to escalate a complaint, then the alternative text could contain a static graphic of the flow chart along with explanatory text of the steps to escalate a complaint. This is but one example of making multimedia accessible. This topic is extensive and is changing rapidly. For a more detailed discussion of making Flash movies accessible, see Smith (2004).

## BARRIER 6: LACK OF LOGICAL TEXT FOR HYPERLINKS

Text used for a hyperlink "should be meaningful enough to make sense when read out of context—either on its own or as part of a sequence of links" (World Wide Web Consortium 1999c). As with text used for an alt attribute, instead of making the actual link a phrase such as "click here," make it a more substantive phrase such as "information about kayaks." Contextual references such as "click here" have little meaning when they are read aloud or when they are presented in a list of links, as they do not provide any clues about what information will be found if a user follows the link. This recommendation is covered under WCAG 1.0 checkpoint 13.1 [Priority 2].
WCAG 1.0 checkpoint 13.1 also recommends using the title attribute within an <a> element to provide further clarification. Nielsen (1998) notes that "such explanations can give users a preview of where the link will lead and improve their navigation: bad links are less likely to be followed; users will waste less time going down the garden path [and] increasing users' understanding of good links helps them interpret the destination page upon arrival: disorientation is reduced."

For sighted users, these titles appear as pop-up text when a cursor hovers over the link. For persons using screen readers, the titles, if activated, will be read in addition to the text of the link. Content designers can also let screen reader users know that the link will open a new window, a strategy that helps persons using ATs keep track of how many browser windows they have open.

## BARRIER 7: MISLEADING USE OF STRUCTURAL ELEMENTS ON PAGES

In HTML 4.0, elements are classified as structure, presentation, metadata, processing, alternative, or replaced elements. In defining the logical organization of a document, the World Wide Web Consortium (2003) notes that "an element (e.g., P, STRONG, BLOCKQUOTE in HTML) that specifies document structure is called a structural element.… [A]n element that specifies document presentation (e.g., B, FONT, CENTER) is called a presentation element."

Content developers are urged not to misuse structural elements to achieve a certain "look and feel" for their page. "Frequent candidates included headings…, used to change font sizes and styles; lists…, used to create a format in which every other line is indented, as in some poems; and block quotations…, used to indent text" (Slatin and Rush 2003, 489). Some content developers tag content with heading elements because they like the way the font and weight of the heading element looks on the screen. A visual user usually does not notice this shortcut. Problems arise when accessibility devices like ATs come into play. Screen readers such as JAWS use the Web page's structure to help users who are visually impaired navigate through the page. Some screen readers even allow users to jump from one heading to another (in JAWS, users can press the H key to move from heading to heading) so that they can get a feel for (or outline of) the information that is contained on that page. Misused heading elements can cause confusion by giving minor content pieces an undeserved importance, and a lack of headings can leave users feeling confused.

Using the <blockquote> or <Q> elements just to indent content instead of a quoted piece of information can also be confusing. These two elements are intended to indent block and inline quotations on a Web page. When the contents of a <blockquote> element are read aloud by a screen reader like JAWS, a user hears the word "blockquote," then the quoted material, and then the words "blockquote end." If the <blockquote> element

 

does not contain an actual quote and the user cannot see that the <blockquote> element was used just for visual purposes, a person using a screen reader could be very confused.

Therefore, the experts (World Wide Web Consortium 2000a; Slatin and Rush 2003; Paciello 2000) advise that when designing a Web site or page, content developers should divorce the structure of the page from the presentation of the page. In other words, do not immediately envision what a Web site will look like; designing with a look in mind may ensure that a Web page will have some built-in accessibility problems. Instead, the World Wide Web Consortium (2000a) suggests creating an outline of the document or, as Paciello (2000, 96) notes, for new content, creating a list of the major components to be included. In this outline, each level denotes a structural change. Use structural elements to mark the structural changes and presentation elements to make them more apparent visually (World Wide Web Consortium 2000a). By identifying which HTML elements are appropriate for each piece of the outline, content developers then will have a structure for their Web page that they can visually present to users in a way that works best for all users—both with and without disabilities.

## BARRIER 8: TABLES THAT ARE DIFFICULT TO DECIPHER WHEN LINEARIZED

Tables are very popular in Web pages and are, as Paciello (2000, 109) states, "some of the most abused elements in HTML." Web designers use tables both to organize content (data tables) and to provide a structure for a Web page (layout tables). This section focuses on providing accessible data tables.

Content developers are urged to think about how their tables will be read (linearized) by an AT. Generally, when a simple data table is linearized, the contents of the cells become a series of paragraphs and are read from left to right and down the table. This can lead to a confusing translation of a table if unrelated items are in cells next to each other.

To combat this issue, content developers should include structural elements so the AT can provide to a user the context of the data table. To provide this context, developers should add to each column in a data table a heading <th> element and within this heading element provide a unique id attribute. Then, developers should link each data table cell in a table to its column header by providing a headers attribute within the table data cell <td> element that contains its associated column ID attribute.

The summary attribute of the <table> element can also be used to provide useful contextual information for people using ATs. This attribute is especially useful when it is used to provide information about the content and organization of the table. The World Wide Web Consortium (2000c) states, "A summary of the relationships among cells is especially important for tables with nested headings, cells that span multiple columns or rows, or other relationships that may not be obvious from analyzing the structure of the table but that may be apparent in a visual rendering of the table." The information in a summary attribute does not appear on the Web page but is read aloud by ATs. The summary attribute is covered under WCAG 1.0 checkpoint 5.5 [Priority 3].

When read aloud by a screen reader like JAWS, the program reads aloud the number of columns and rows the table contains, the contents of the summary attribute (if it has been provided), and the table data. Using a combination of key strokes allows a JAWS user to obtain the header information for a cell. Having the ability to retrieve this contextual information makes this table much easier to decipher than the previous one.

## BARRIER 9: SITES WITH POOR COLOR CONTRAST

Color blindness is a condition in which a person has faulty or missing color detectors (cone cells) and has "trouble distinguishing between combinations and/or pairs of colors" (Paciello 2000, 8). About one in 10 U.S. males is color blind; color blindness, however, rarely affects women. The most common type of color blindness creates a problem in distinguishing some shades of red and green. WCAG 1.0 checkpoint 2.2 states, "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. [Priority 2 for images, Priority 3 for text]" (World Wide Web Consortium 1999c).

To analyze a Web page for contrast, the World Wide Web Consortium (2000b) suggests looking at it using a monochrome monitor or using a browser that has colors turned off. If those options are not possible, they also suggest printing the Web pages on a black and white printer (with backgrounds and colors appearing in grayscale) to see how the color scheme looks to people with color deficiencies or with low-resolution monitors. Content designers can also to test their pages with tools such as Vischeck (http://www.vischeck.com), which

simulates what a Web page will look like to people who are color blind.

## BARRIER 10: LACK OF TESTING

Once their Web pages have been built, developers should test their Web pages. Tests for compliance with the WCAG 1.0 guidelines can be done using free online tools such as WebXACT (http://webxact.watchfire.com) or World Wide Web Consortium Markup Validation Service (http://validator.w3.org/). These tools analyze whether or not Web pages comply with the guidelines and note any accessibility issues.

Another test is for a content developer to think about how a page will be presented to a person with disabilities. The reading order of the headers, images, links, and content of a Web page makes a large impact on the usability of the page. In general, ATs read or display information linearly from the left to the right of the page and from top to the bottom of the page. Tools such as WAVE (http://www.wave.webaim.org) can help a content designer to evaluate a single Web page's reading order. On WAVE's main page, the user can enter the address of the Web page to be evaluated. WAVE then directs the user to a feedback page so that he or she can evaluate the order in which the Web page is read as well as receive feedback on any other accessibility guideline issues the Web page may have.

But a designer should not rely on the tools alone. Invite several people with one or more disabilities to use the Web site. These users can tell the designer where they are having real difficulties with finding information in a Web site. As Slatin and Rush (2003, 15) say, "Hearing and seeing and feeling your Web site through the ears and eyes and hands of people with disabilities can be a surprising and sobering experience." A designer may find that he or she has complied with the guidelines to the letter but has not provided a meaningful experience to a person with impaired vision, for example, or to someone who cannot hear the audio elements.

## CONCLUSION

Content designers should think about accessibility throughout the design of their Web site. During the design process, content developers should plan to add to their Web pages the HTML attributes and elements to support the WCAG 1.0 guidelines, some of which have been discussed in this article. Content developers should also test their Web pages for compliance with the accessibility guidelines and invite people with disabilities to test their pages so they can experience real users navigating their site. Employing these techniques can help content designers deal with accessibility issues before their online information goes live and may help to reduce rework and redesign.

## REFERENCES


Nielsen, J. 1998. Using link titles to help users predict where they are going. http://www.useit.com/alertbox/980111.html.

———. 2000. Designing Web accessibility. Indianapolis, IN: New Riders Publishing.

Paciello, M. G. 2000. Web accessibility for people with disabilities. Lawrence, KS: CMP Books.

Slatin, J. M. 2001. The art of ALT: Toward a more accessible Web. Computers and composition 18:73-81.

Slatin, J. M., and S. Rush. 2003. Maximum accessibility. Boston, MA: Addison-Wesley.

Smith, J. 2004. Creating accessible Macromedia Flash content. http://www.webaim.org/techniques/flash/.

Theofanos, M. F., and J. C. Redish. 2003. Guidelines for accessible and usable Web sites: Observing users who work with screen readers. Interactions 10 (6): 36-51.

WebAim. 2005. Creating accessible frames. http://www.webaim.org/techniques/frames.

World Wide Web Consortium. 1999a. Fact sheet for Web Content Accessibility Guidelines 1.0. http://www.w3.org/1999/05/WCAG-REC-fact.

———. 1999b. Techniques for Web Content Accessibility Guidelines 1.0. http://www.w3.org/TR/1999/WAI-WEBCONTENT-TECHS-19990505/wai-pageauth-tech.html.

———. 1999c. Web Content Accessibility Guidelines 1.0. http://www.w3.org/TR/WAI-WEBCONTENT/.

———. 2000a. Core techniques for Web Content Accessibility Guidelines 1.0. http://www.w3.org/TR/WCAG10-CORE-TECHS/.

———. 2000b. CSS techniques for Web Content Accessibility Guidelines 1.0. http://www.w3.org/TR/WCAG10-CSS-TECHS/.

———. 2000c. HTML techniques for Web Content Accessibility Guidelines 1.0. http://www.w3.org/TR/WCAG10-HTML-TECHS/.

———. 2001. WAI quick tips reference card. http://www.w3.org/WAI/References/QuickTips/.

———. 2003. Glossary of Web Content Accessibility Guidelines 1.0. http://www.w3.org/2003/glossary/subglossary/wcag10.rdf/.



Linda Enders Roberts
Technical Writer
SAS Institute Inc.
Cary, NC 27713


Linda Enders Roberts has worked for more than 15 years as a technical writer, trainer, and project manager.

 