Technical Assistance Manual

Section 508

Website Accessibility



District of Columbia

Office of Disability Rights

Last updated 6/11/13

**Introduction**

This manual provides basic guidance for making web content developed for the Government of the District of Columbia accessible to persons with disabilities. It offers techniques for developing and testing information and interfaces developed for web and social media sites so they can be accessed by all users. Information for using websites and social media to reach and provide information to citizens with disabilities is also included.

The varieties of methods and formats for using the web to deliver information is almost limitless. For this reason, this guide will not present solutions for every way information can be delivered, but rather offer basic concepts and techniques for making information accessible. Developers, authors and editors should use the information contained in this guide as a starting point to learn about web accessibility, and then go on to explore additional resources to obtain techniques for achieving accessibility within their specific web environment. Where possible, resources for common web environments and formats used throughout the D.C. Government are provided in this guide and should be consulted for further information.

**Table of Contents**

[Definition of Terms and Acronyms 7](#_Toc4497620)

[Accessibility Overview 9](#_Toc4497621)

[Disability Types 10](#_Toc4497622)

[Accessibility Standards 12](#_Toc4497623)

[Section 508 12](#_Toc4497624)

[Web Content Accessibility Guidelines 13](#_Toc4497625)

[Assistive Technologies 13](#_Toc4497626)

[Screen Readers 14](#_Toc4497627)

[Screen Magnifiers 14](#_Toc4497628)

[Speech Recognition Software 15](#_Toc4497629)

[Web Accessibility Requirements 15](#_Toc4497630)

[Images 16](#_Toc4497631)

[Requirements 16](#_Toc4497632)

[Techniques 18](#_Toc4497633)

[How to Test 18](#_Toc4497634)

[Color 19](#_Toc4497635)

[Requirements 19](#_Toc4497636)

[Techniques 20](#_Toc4497637)

[How to Test 20](#_Toc4497638)

[Tables 21](#_Toc4497639)

[Requirements 21](#_Toc4497640)

[Techniques 22](#_Toc4497641)

[How to Test 23](#_Toc4497642)

[Forms 23](#_Toc4497643)

[Requirements 24](#_Toc4497644)

[Techniques 26](#_Toc4497645)

[How to Test 26](#_Toc4497646)

[Links 27](#_Toc4497647)

[Requirements 28](#_Toc4497648)

[Techniques 29](#_Toc4497649)

[How to Test 30](#_Toc4497650)

[Frames 31](#_Toc4497651)

[Requirements 31](#_Toc4497652)

[Techniques 32](#_Toc4497653)

[How to Test 32](#_Toc4497654)

[Page Structure 32](#_Toc4497655)

[Requirements 33](#_Toc4497656)

[Techniques 33](#_Toc4497657)

[How to Test 34](#_Toc4497658)

[Lists 34](#_Toc4497659)

[Requirements 35](#_Toc4497660)

[Techniques 35](#_Toc4497661)

[How to Test 36](#_Toc4497662)

[Style Sheets and Styles 36](#_Toc4497663)

[Requirements 37](#_Toc4497664)

[Techniques 37](#_Toc4497665)

[How to Test 38](#_Toc4497666)

[Embedded Content 38](#_Toc4497667)

[Requirements 39](#_Toc4497668)

[Techniques 40](#_Toc4497669)

[How to Test 40](#_Toc4497670)

[Animation 41](#_Toc4497671)

[Requirements 41](#_Toc4497672)

[Techniques 42](#_Toc4497673)

[How to Test 42](#_Toc4497674)

[JavaScript 42](#_Toc4497675)

[Requirements 43](#_Toc4497676)

[Techniques 43](#_Toc4497677)

[How to Test 44](#_Toc4497678)

[WAI-ARIA 45](#_Toc4497679)

[Requirements 46](#_Toc4497680)

[Techniques 46](#_Toc4497681)

[How to Test 46](#_Toc4497682)

[The following MSAA properties can be tested with Object Inspector : 47](#_Toc4497683)

[References and Tools 47](#_Toc4497684)

[Color Testing: 48](#_Toc4497685)

[MSAA Testing Tools: 48](#_Toc4497686)

[PDF Accessibility Tools: 48](#_Toc4497687)

[Section 508 Guide and Best Practices: 48](#_Toc4497688)

[Screen Readers: 48](#_Toc4497689)

[Data Tables: 48](#_Toc4497690)

[Social Media Accessibility 49](#_Toc4497691)

[Overview 49](#_Toc4497692)

[Social Media Networks 49](#_Toc4497693)

[Facebook 50](#_Toc4497694)

[Twitter 51](#_Toc4497695)

[YouTube 52](#_Toc4497696)

[User Impact 52](#_Toc4497697)

[Requirements 54](#_Toc4497698)

[Reaching Constituents with Disabilities 57](#_Toc4497699)

[Using Websites for Outreach 57](#_Toc4497700)

[Using Social Media for Outreach 58](#_Toc4497701)

[Using Blogs and Newsletters 59](#_Toc4497702)

[Providing Alternate Formats and Methods of Presentation 60](#_Toc4497703)

[Using District and Disability Resources 60](#_Toc4497704)

[Appendix A – Checklists 61](#_Toc4497705)

# Definition of Terms and Acronyms

A

API: Application Programming Interface - A set of code and specifications that programs can interface to communicate with each other to perform an action

AT: Assistive Technology - A term that includes assistive, adaptive, and rehabilitative devices for people with disabilities that promote greater independence by enabling people to perform tasks that they were formerly unable to accomplish, or had great difficulty accomplishing, by providing enhancements to or changed methods of interacting with the technology needed to accomplish such tasks

AJAX: Asynchronous JavaScript and XML - A method of exchanging data with a server and updating contents of a web page or application without reloading the whole page

Alternative Text: Attribute of an image that incorporates text in order to describe the image to users of screen readers. When an image is meaningful to the page, then the alternative text must be descriptive and informative. If the image is purely decorative, then assistive technologies must be informed to ignore it through the use of null alternative text.

ARIA: Accessible Rich Internet Application - A technical specification that provides a framework to improve the accessibility and interoperability of web content and applications developed with Ajax, HTML, JavaScript and related technologies

F

Frames: Frames is the use of multiple, independently controllable sections on a Web presentation. This effect is achieved by building each section as a separate HTML file and having one "master" HTML file identify all of the sections.

G

Graceful Degradation: The property that enables a system (in this instance, computer-based) to continue operating properly in the event of the failure of some of its components (e.g. ARIA not supported with particular browser and assistive technology combinations)

H

HTML: Hypertext Markup Language

I

IE: Internet Explorer (browser)

IFrame: An IFrame (Inline Frame) is an HTML document embedded inside another HTML document on a website. The IFrame HTML element is often used to insert content from another source, such as an advertisement, into a Web page.

K

Keyboard Focus: The location where keyboard actions will be interpreted by the application

L

List (ordered): A group of list items that convey a hierarchal relationship based on numbers or letters.

List (unordered): A group of list items that lack a hierarchal relationship, and are presented to the user using bullets as opposed to ordered numbers or letters.

List Item: Blocks of text or content that is grouped with other related content placed in a list to relay to the user an explicit relationship.

M

MSAA: Microsoft Active Accessibility - An accessibility API that provides adaptive technology users the role, name, value, and state of user interface components

O

Object: Any entity that can be manipulated by the commands of a programming language, such as a value, variable, function, or data structure.

R

RIA: Rich Internet Application - A web application that has many of the characteristics of a desktop software application including dynamic page elements and dynamic content updates not typically implemented solely with HTML

S

Screen Reader: Software designed to audibly render content to non-sighted users based on source code, PDF tag tree and document content.

Style Sheets: External documents that can be applied to a web page to style, position and insert content into the webpage.

W

WAI: Web Accessibility Initiative - Group of W3C members that develop strategies, guidelines, and resources to improve Web accessibility to people with disabilities

WAI-ARIA: Web Accessibility Initiative - Accessible Rich Internet Applications

W3C: World Wide Web Consortium - International standards organization for the World Wide Web

# Accessibility Overview

Introduction

As D.C. government departments transition from delivering programs and services via traditional paper-based formats to the use of web-based technologies, it becomes important to make sure these programs and services are available to all citizens regardless of their abilities and the methods they use to access the web. This is the subject of web accessibility. While the standard definition of accessibility concerns the degree to which information, services and the physical environment is available to people with different types of disabilities, the broader topic addresses the ability to access information, programs and services regardless of the device, method or mode of presentation being used. Whether accessing a web page via a computer, tablet, mobile phone or other device, web pages should scale to provide an equal browsing experience regardless of device used. Users should be able to control web interfaces regardless of whether they are using pointing devices such as a mouse, keyboards and keyboard emulators, touch screens or speech recognition software. Users should be able to understand information on web pages if they need to change text color or text size, read the web page using speech synthesis or refreshable Braille, if they need captions for audio content or if they require language that is easier to understand.

The topic of universal design covers making technology available to all users in as many contexts as possible. The web site www.usability.gov provides additional information on the topics of usability and universal design. Site developers and designers should strive to use universal design principles as much as possible. The DC.Gov Accessibility Policy implements the accessibility standards which federal government agencies are required to meet to make their technology accessible to people with disabilities. These standards are contained in Section 508 of the Rehabilitation Act of 1998. The Section 508 Electronic and Information Technology (EIT) standards contain requirements which must be met for all web pages on the DC.Gov portal to ensure they can be accessed by users with disabilities. The District of Columbia is not a federal government establishment, so Section 508 does not apply directly to the D.C. government. Title II of the Americans with Disabilities Act of 1990 (ADA), however, requires all state and local governments to take steps to make sure that the communications they provide to people with disabilities are as effective as the communications provided to others . (This requirement to provide “effective communication” also applies to private establishments under Title III of the Americans with Disabilities Act.) To be certain that all web and technology communications are as effective to people with disabilities as they are to others, the D.C. government has adopted the Section 508 requirements as a minimum standard that must be met for all of its web communications.

# Disability Types

Over 56 million Americans have some type of disability, among them growing numbers of veterans. According to the 2010 U.S. Census, one out of every five Americans has a disability, half of which are considered severe . It is estimated that 1 out of 20 residents of Washington, D.C. ages 18-64 have a disability which has an impact on their employment . The federal Electronic and Information Technology standards adopted by the D.C. government address the functional requirements of users who are blind, low vision, deaf, hard of hearing, speech impaired and have limited movement. This list does not cover all disability types in existence and websites on the DC.Gov portal should be designed to meet the needs of as many users as possible, however the needs of the above disability types must be met as a minimum standard.

Vision

Individuals with visual disabilities lack the ability or have difficulty with sight. Visual disabilities span a wide variety of disability types, including blindness, reduction of visual acuity (visual range), color blindness and tunnel vision. The adaptations used on the web by people facing visual challenges varies widely based on the user’s needs. Some individuals use third-party software to enlarge text or alter the colors used on the screen. Others may adjust settings built into their computer’s operating system or browser, or they may employ a hardware solution such as a larger monitor or a closed circuit television (CCTV). Individuals with more profound vision loss such as blindness may use screen reading software which converts information that is sent to the screen into speech synthesis or refreshable Braille.

Mobility

Individuals with mobility disabilities have some limitation of movement. These users may control their computer using the keyboard instead of a pointing device such as a mouse, or they may use a hardware solution which emulates a keyboard or pointing devices such as a head pointer. Others may find a software solution to be more effective such as an on-screen keyboard or speech recognition software.

Speech

Individuals with speech disabilities lack the ability or have difficulty producing speech. They often use augmentative communication devices which can range from picture boards to complex speech synthesis systems. Voice communication may also be facilitated through a teletype (TTY) or video relay device.

Auditory

Individuals with auditory disabilities lack the ability or may have difficulty hearing. Individuals with hearing loss may use hearing aids or assistive listening devices such as telecoils or neck loops. Individuals with more profound hearing loss such as deafness may use sign language, captioned audio, TTY or video relay devices to facilitate communication.

Cognitive

Cognitive disabilities are neurological disorders that affect the cognitive process. Cognitive disabilities range a wide variety of disability types, including intellectual disabilities, learning disabilities, behavioral disorders, and autism spectrum disorders. Some individuals with cognitive disabilities benefit from reading systems which use a combination of speech synthesis and page tracking features. Others may use sites such as InstaPaper and Readability to simplify page layout and remove distractions on the page.

Age-Related

In 2011, the U.S. Census published that 37 percent of people 65 and older have at least one disability. The incidence rate for disability increases as people age. Web use and computer penetration among the population age 65 and older is growing. Today the “baby boomer” generation makes heavy use of the web.

# Accessibility Standards

Many countries have laws which uphold the rights of citizens with disabilities. The United States has passed several federal laws which address disability rights and which address access to technology and the physical environment. Many states within the U.S. have also passed similar regulations which address disability rights and access.

# Section 508

Section 508 is a U.S. federal law requiring that electronic and information technology that is developed, purchased, used or maintained by the federal government is accessible to people with disabilities. Section 508 adds accessibility requirements to technology purchases by U.S. federal agencies. While Section 508 only directly applies to federal agencies, many state governments have adopted the Section 508 requirements at the state level. The government of the District of Columbia applies the Section 508 standards to information that is posted on the DC.Gov web portal.

The Section 508 standards are broken down into categories. The four main categories are the general requirements, technical standards, functional performance criteria, and information, documentation and support requirements. The technology categories covered under the technical standards include software applications and operating systems, web-based information and applications, communications products such as telephone and voice mail systems, multimedia and video products, self-contained systems such as kiosks and office equipment, and computer systems.

# Web Content Accessibility Guidelines

The Web Content Accessibility Guidelines (WCAG) are a set of international standards created by the Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C). They are not legal requirements but rather a set of international standards which are often used as the basis for domestic web accessibility regulations and standards in many countries. The current Section 508 standards are based on version 1.0 of the Web Content Accessibility Guidelines which were adopted in 1999. In 2008, WCAG 2.0 was adopted by the W3C to address advancements in web technologies. WCAG 2.0 is divided into three levels that content creators should meet to conform to WCAG 2.0: level A, level AA and level AAA. Level A comprises the minimum framework that a site must meet in order to achieve accessibility, whereas level AAA contains all WCAG 2.0 requirements.

When creating websites, developers should target to meet the requirements of WCAG 2.0 level AA conformance beyond the Section 508 requirements. There are two reasons for this. The WCAG 2.0 criteria are more detailed than the Section 508 requirements and therefore likely to produce a more accessible experience for users with disabilities. Additionally, as of this writing, the Section 508 standards are being updated to reflect changes in technology since the original standards were passed in 2001. The updated standards being proposed by the U.S. Access Board are based on the WCAG 2.0 level AA criteria, and creating a site that meets these requirements will prepare developers when the Section 508 standards are updated. While meeting the WCAG 2.0 AAA criteria is the ideal scenario, the W3C even acknowledges that it is unrealistic for every page of a website to meet all of the WCAG 2.0 level AAA requirements.

# Assistive Technologies

Assistive technologies are devices, software or techniques used to assist individuals with disabilities in the performance of tasks which might otherwise be difficult or impossible. Examples of assistive technologies include screen readers and magnifiers, on-screen or other special keyboards, wheelchairs, hearing aids, TTY devices, canes, and refreshable Braille displays.

# Screen Readers

Screen readers function by reading the contents of the computer screen aloud using speech synthesis and optionally an attached refreshable Braille display. They provide keyboard-based input commands to replace visually dependent input devices such as a mouse. Examples of screen reader programs include JAWS for Windows by Freedom Scientific, Window-Eyes by GW Micro, System Access by Serotek, Nonvisual Desktop Access (NVDA) by NV Access, Microsoft Narrator (built into Windows 2000 and later), and VoiceOver (built into Mac OS 10.4 or higher and iOS 3.5 or higher). WebAIM has published a survey detailing screen reader usage statistics. The most recent survey as of this writing can be accessed by going to http://webaim.org/projects/screenreadersurvey4.

Screen readers provide multiple ways for users to access information on a web page. Users can review page content using the arrow keys as if the page was a word processing document, or they can tab to all active links and controls on the page. Many screen readers also provide navigation keystrokes to jump to certain control types such as headings or lists, and many will also display all links and controls on the page in a list for easy review.

# Screen Magnifiers

Screen magnifiers enlarge contents of the screen, allowing users with low vision to read printed information and see images and other information clearly. Magnification levels from 1.2 to 16 times are typically supported. Image and color manipulation features are often included to make images easier to see. Limited screen reading functionality via speech synthesis is often included. Other features provided by screen magnifiers include the ability to change screen colors and to enlarge the system caret or mouse pointer. Examples of screen magnifiers include ZoomText from Ai Squared, MAGic from Freedom Scientific, SuperNova from Dolphin Computer Access, Magnifier (included with Windows 2000 and later), and Zoom (included with MacOS 10.4 and higher and iOS 3.5 and higher).

# Speech Recognition Software

Speech recognition software allows users with limited mobility to control the computer using their voice. At a basic level, speech recognition packages allow users to dictate text and enter it into an application as if typed using the keyboard. More advanced speech recognition applications allow the user to issue commands to be carried out by the current application or the operating system. These can include commands to replace keystrokes and mouse actions. Common speech recognition programs include Dragon from Nuance Communications, Windows Speech Recognition (built into Windows Vista and later), and Dictation and Speakable Items (built into Mac OS Mountain Lion).

Speech recognition allows the user to speak the name of a link or control in order to access it. Users can also call up a list of links or related controls and speak the adjacent number to select the desired one. Users can dictate text into text boxes, or speak commands to send keystrokes to the application or move the mouse pointer.

# Web Accessibility Requirements

This section outlines basic requirements for achieving compliance with Section 508 and making web information accessible to people with disabilities. The information is divided into general areas of accessibility practice. For example, the “Images” section provides information on issues and techniques relating to images. Each section contains a description of issues including user impact, requirements for resolving the issues, and different evaluation techniques. Resources are provided for achieving compliance with each requirement using HTML, Adobe Acrobat PDF and Adobe Flash. Readers should refer to OCTO documentation for achieving compliance using the DC.Gov Drupal CMS environment, and to other resources to gain a full list of Section 508 requirements and techniques for designing, developing and testing accessible web content.

# Images

Images are pictures which are embedded in a document. For most web pages, the image content is contained in an external file and then embedded in the page by referencing the file location within the web page, causing it to be displayed as part of the page by the browser. The file only contains information required to form the shapes and colors which make up the image, and as such does not contain any text that could be interpreted by a computer, browser or assistive technology. If images are used to provide meaning or information which contributes to the understanding of the page, users who cannot see the image will lose the meaning that the image represents if it is not described. The foreground and background colors used to make up the image may not provide enough contrast for it to be clearly seen by a user with low vision. If the image contains written text, the user will be unable to use magnification to enlarge the text within the image, or the image may become pixelated (visually broken into separate dots). If a background image is displayed using cascading style sheets (CSS), the image will disappear if the user turns on the Windows high contrast mode which reverses the colors used on the screen. If an image is part of a link, users of voice recognition software will speak the alternative text provided for an image to gain access to the link. For these reasons, it is important that alternatives are provided to identify images for the benefit of those who are unable to see them.

Requirements

The first step is to determine if an image contributes to the understanding of the page. Images fall into two categories: “meaningful” and “decorative.” Meaningful images provide information that is not otherwise available on the page. Decorative images are used for visual effect, or they provide information that is also available on the page as text. For meaningful images, a text alternative which conveys the same meaning indicated by the image must be provided. For decorative images, assistive technologies should be informed to ignore the image so that confusing or extraneous information will not be provided about it.

Concise, informative alternative text should be provided for meaningful images. The goal of alternative text should be to present text which will provide the same level of understanding to those who cannot see the image as it does to those who can. Alternative text should represent the purpose of the image rather than a description of its visual appearance. For example, if an image of a printer is used to represent the function of printing the web page, the alternative text should be “print this page” instead of “picture of a printer.” If an image of a progress bar is used to indicate the amount of completion for a task, alternative text of “75 percent complete” is more useful than alternative text of “progress bar.” Screen readers will usually announce when an image has been encountered, so the words “image,” “picture” and “photo” should not be included as part of the image alternative text. Complex images such as charts, diagrams and graphs will require more lengthy descriptions to gain a full understanding of the image. These descriptions will likely need to appear on a separate page or elsewhere on the same page and should be visible so that users who have difficulty understanding or seeing the images can benefit from the descriptions.

Decorative images are those that are used for visual effect or that do not lend any meaning to an understanding of the page content. Examples of decorative images include lines, spacers, and watermarks. Images should also be considered decorative if the same information is provided using text on the page. In any of these cases, assistive technologies should be instructed to ignore the image so that no information is provided about it. The method for marking an image as decorative varies between document types. For web pages, this is achieved by providing empty or “null” alternative text. Images are deemed decorative in PDF documents and Flash applications by hiding them from assistive technologies. It is important that if an image does not contain useful information, it is properly marked as a decorative image so that screen readers and other assistive technologies will not try to provide information about it by “guessing” as to the meaning of the image. Including a decorative image in a document without providing the proper steps to ignore it will result in information being announced by assistive technology about the image to assist the user in interpreting the meaning of the image, such as the file name or server location of the image. This information can often be meaningless and will likely leave the user confused as to the purpose of the image.

An image map is an image containing multiple active “regions,” or areas, inside the image which perform separate functions or go to different destinations. An example of an image map would be a map of a country, where each region was a state within the country that took the user to a different location on the website. There are two types of image maps: server-side and client-side image maps. The individual regions of server-side image maps respond to mouse clicks and cannot be reached using the keyboard. For this reason, client-side image maps should always be used in favor of server-side image maps. Where server-side image maps must be used, alternatives would need to be provided on the page (such as separate links which respond to the keyboard).

Techniques

Providing text alternatives for images will vary depending on the authoring platform and the image type. For information on providing text alternatives for images on web pages in the HTML source code, refer to the Text Tags section of the “Guide to the Standards” document on the U.S. Access Board web site. For additional information, refer to the Images best practices section of the Social Security Administration Best Practices Library, which includes techniques for providing alternative text, methods for testing image alternatives, and an “Alternative Text Guide” for writing effective alternative text. For information on making image maps accessible, refer to the Server-Side Image Maps and the Client-Side Image Maps section of the “Guide to the Standards” document on the U.S. Access Board website. For techniques to provide image alternatives in Adobe Acrobat PDF documents, refer to the Providing Alternative Text for Images and the Hiding or "Artifacting" Non-Meaningful Content sections of the “Creating Accessible PDFs Tutorial” on the United States Department of Veterans Affairs website. To learn how to provide text equivalents within Adobe Flash applications, refer to the Providing Text Equivalents and Hiding Flash sections of the “Creating Accessible Flash course” on the United States Department of Veterans Affairs website.

How to Test

There are several ways to verify that images are accessible. One of the best ways to test this is to read the page using a screen reader, ensuring that alternative text is provided for images and that image maps can be accessed using the keyboard. Some screen readers provide a command to bring up a list of all images on the page. This not only allows for quick checking of all images at once, but has the added benefit of visually displaying on the screen what will be announced by the screen reader for each image when it is encountered. For information on how to use JAWS for Windows to access images, refer to the Navigating Web Pages section of the “Surf’s Up!” tutorial on the Freedom Scientific website. Another method for checking image alternatives is to use a tool that will scan web pages and PDF documents for errors in the page source code. These tools can evaluate large numbers of pages very quickly to catch images which are missing meaningful text alternatives, but the page will still need to be reviewed by a human to ensure that any provided image alternatives present the same level of understanding as the image. Some browsers such as Internet Explorer will display the alternative text of the image as a tooltip when hovering over the image using a mouse. Finally, image alternatives can be verified by human review of the page source code. This task can be assisted by the use of toolbars which can identify images and display any provide alternative text.

# Color

Related to the use of images on web pages is the use of color. Often color will be used to indicate the state of items on the page, such as identifying items which have been selected, or form fields which need to be corrected due to errors. Color can also be used to tell items apart such as in the case of a chart or graph where a legend or key relies on color. Users who are blind will not receive the information provided by the use of color. Users with colorblindness may be unable to tell the colors apart used on the page. Users with low vision may be unable to tell the difference between the foreground and background colors used on the page which makes content difficult to see. For these reasons, color should be used in a way which does not prevent users from obtaining the information necessary to gaining an understanding the page.

Requirements

When color is used to identify elements or controls on the screen, or if color is used to give instructions, an alternate method must be included to provide the same information. This is also true for the use of changes in font characteristics and other forms of visual formatting. When visual formatting is used, users who are blind, colorblind, low vision or who use black-and-white monitors may miss the changes in visual formatting which may prevent them from being able to use the web page. This requirement does not prevent the use of color to enhance the identification of important features. It does, however, require that some other method of identification be combined with the use of color. Color or other visual formatting which is solely used as decoration or for visual appeal does not apply to this requirement.

Color contrast refers to the degree of difference between the foreground and background colors used on the page. The foreground color is the color used to display text, images, controls and other page content. The background color is the color against which items on the page are presented. When the page foreground and background colors are too much alike, users with low vision or color deficiencies may have difficulty understanding page content. While many applications allow users to change page colors, users will be unable to alter the color of text which is included as part of an image. Color pairings for text and for images which contain text must contain enough difference between foreground and background colors to allow content to be easily seen. The contrast between foreground and background colors is stated as the “color contrast ratio.” While the current Section 508 standards do not include color contrast requirements, minimum color contrast ratios are contained as part of the Web Content Accessibility Guidelines 2.0, the proposed requirement that will likely be adopted when the Section 508 standards are updated. According to the WCAG 2.0 requirement, text which uses less than an 18-point font or which uses less than a 14-point bolded font must use a color contrast ratio of 4.5 to 1 or more. Text which uses an 18-point or larger font, or text which use a 14-point or larger bolded font must have a color contrast ratio of 3 to 1 or more. For more information on this requirement, refer to Understanding WCAG 2.0 Success Criterion 1.4.3 on the W3C-WAI WCAG 2.0 site.

Techniques

The method for supplementing information which is provided using color is to provide another method of identification which uses text. This text can be provided visually on the page, or it can be provided in a manner that cannot be detected visually but can be detected by assistive technologies. The preferred method is to make the text visible so that it will benefit individuals who have low vision or color deficiencies. For more on supplementing information that is provided using color, refer to the Color section of the “Guide to the Standards” on the U.S. Access Board website.

Templates for web pages and other documents should follow color and visual formatting requirements prescribed for the DC.Gov web portal. This also includes ensuring that foreground and background colors provide a high degree of color contrast. Techniques to achieve proper color contrast requirements can be accessed by visiting How to Meet Success Criterion 1.4.3 on the W3C-WAI WCAG 2.0 website. Techniques for the accessible use of color within Adobe Acrobat PDF documents can be found in the Using Color in an Accessible Way section of the “Creating Accessible PDFs Tutorial” on the U.S. Department of Veterans Affairs website. Techniques for the accessible use of color within Adobe Flash applications can be found in the Using Color section of the “Creating Accessible Flash Course” on the U.S. Department of Veterans Affairs website.

How to Test

One way to ensure that the website does not rely on the ability to see color is by reviewing it using a screen reader. If the page can be fully understood using the screen reader alone, then color is not preventing the understanding of the web page. Similar techniques include viewing the website using a black-and-white monitor or printing the web pages as black-and-white text. A quick way to remove color from a web page is to submit the web page address to the GrayBit Grayscale Conversion Contrast Accessibility Tool. Color can be verified to not prevent understanding of a web page if the page can be fully understood in the absence of all color.

Color contrast can be tested by using an eyedropper tool to obtain the numeric values that represent the foreground and background colors on a web page to compute the color contrast ratio. Several tools exist to aid with color contrast ratio calculation. One such tool is the Contrast Analyzer for Windows and Mac from the Web Accessibility Tools Consortium and the Paciello Group. This tool is an application which must be downloaded and installed on the computer being used for testing. Several online color contrast analyzers will calculate the contrast ratio when the numeric color values are submitted. These tools include the Color Contrast Check Tool on the snook.ca website, and the SSB Color Contrast Checker.

# Tables

Tables visually display relationships between pieces of information using formatted rows and columns. These types of tables are referred to as data tables because they contain either row or column headers which identify how all the data in a particular row or column is related. Tables which organize information into rows and columns in order to present a particular visual appearance but which do not use tables to organize related data are called layout tables, and according to OCTO requirements, are not permitted. Visual appearance and presentation of information should be controlled using Cascading Style Sheets (CSS) and not using data table coding.

When data tables are not coded correctly, users of assistive technology will have difficulty moving around the table and understanding relationships between data within the table. Consider the example of a schedule for TV programs which displays the listings as a grid. Each column in the grid represents TV channels for which programs are displayed. Each row In the grid represents times at which each program airs. The cells of the grid represent the individual programs that are listed. Without a properly coded data table, a user would be unable to move to each program (cell) within the grid and determine the channel (column header) and time (row header) in order to access the program.

Requirements

Information which is related and laid out as a grid containing rows and columns should be coded as a data table. Visually creating rows and columns using spacing, tab stops, and other methods used to visually create rows and columns will not allow users of assistive technology to review the table or effectively move between cells, rows and columns within the table. Properly coded tables means that the coding rules are followed that allow computers to correctly identify individual tables, rows, columns, cells, column headers and row headers. When this information is provided via code, it can be provided to the user using assistive technology, and users can move around the table by cell, row or column. Properly coding tables also means that barriers have been removed from the table which would prevent review, movement or understanding. For example, placing tables inside each other (nesting), or merging cells together can create difficulty moving between cells or maintaining the relationship between the cells and the row or column headers. In a similar fashion, splitting tables apart so that the row and column headers and the cells are not in the same table will cause users of assistive technology to lose this relationship as well.

Table row and column headers must be labeled as such using coding techniques. When row and column headers use bolding, larger fonts or other changes in visual appearance, they will not be identified as headers by assistive technology which will increase the difficulty in understanding the purpose of each table cell. This also means that coding techniques used to identify row and column headers should not be used on table cells which are not actually table headers. When row or column headers are labeled using coding techniques and associated with the table cells that they label, screen readers will announce each header as the user moves between rows or between columns. Table headers and cells must be located in the same table and not separated by any barriers such as additional tables in order for the relationships to be announced by screen readers.

Additional information about a table should be provided if doing so would help to better understand the information or layout of the table. Providing a caption for a table can help to identify each table on the page or to explain the information contained in the table. Captions appear visually so they will not only benefit users of screen readers, but also users with low vision and cognitive disabilities. Table summaries should be used to provide information to users of screen readers about the visual layout of the table, as table summaries are not visually shown.

Techniques

Data table accessibility requires using the proper coding rules to identify tables, rows, columns, table cells, and row and column headers. For information on making tables accessible in web pages, refer to the Data Table section of the “Guide to the Standards” on the U.S. Access Board website. Additional information can be obtained by reviewing the Basic Data Tables best practice which is part of the SSA Accessibility Best Practices Library on the U.S. Social Security Administration website. Information about making data tables accessible in Adobe Acrobat PDF documents can be found in the Constructing accessible Tables section of the “Creating accessible PDFS Tutorial” on the U.S. Department of Veterans Affairs website. Adobe Flash does not provide a method for making table structures accessible to assistive technologies, therefore text alternatives for tables will need to be provided which describe the data in a meaningful fashion.

How to Test

Tools can automatically scan web pages and PDF documents to verify that tables, rows, columns, cells, and row or column headers have been defined according to coding rules. These tools will catch if any of these items are missing but will require human review to verify that they are being used correctly. One tool which can be used to check HTML data tables is the HTML Table Validator on the Web Experience Toolkit site. Another useful way to test for data table accessibility is to review the table using a screen reader. Be advised, however, that some screen readers will try to provide access to incorrectly coded data tables for users with disabilities by assuming that headers are in the first row or left column, announcing these cell contents as headers. Many screen readers have the ability to disable this feature so that only headers which are coded as such are announced. To learn how to use JAWS for Windows to review and move around tables, refer to the Tables section of the “Surf’s Up!” training on the Freedom Scientific website. Finally, table code should undergo human review to verify that the table has been correctly defined and is coded to standards. This will require inspection of each element which makes up the table to confirm it is coded correctly.

# Forms

The ability for all users to successfully fill out online forms is critical to ensuring access to the functionality of many websites and web applications. Users with disabilities can face major challenges in filling out and submitting forms which are not properly constructed. For example, users of assistive technology and users who enlarge the screen may have difficulty locating form instructions if they are not placed at the beginning of the form. Poorly designed forms can also make it difficult for users of assistive technology to locate and move to form controls, determine what should be entered into each control and how the information should be formatted, determine which form fields are required, and recover from errors. If the form imposes a time limit and the user has not submitted their information before time runs out, the user’s work may be lost. For these reasons, it is important that barriers are not caused which would prevent users from completing and submitting online forms.

Requirements

Form instructions should be placed at the beginning of a form. If placed in another location such as after the form controls, they may be missed or difficult to locate by users with disabilities. Form instructions should be clear and describe any important information needed to complete the form, such as how fields which are required will be identified on the form.

Form controls should be arranged in a meaningful order. This not only applies to the order in which the form is read, but to the order in which users move between form controls. When the order for reading or moving through the form does not make sense, users are likely to get confused and may think that they missed a form control. Examples of form controls that would be considered out of sequence would include fields that are out of order such as requesting the person’s middle name before requesting their first name, or inserting an unrelated field in the middle of related fields such as a credit card field appearing in the middle of a group of mailing address fields.

Every form control should be associated with a label. This requires using the proper coding techniques to tie the form control and the label that it represents together. When form controls are tied to their respective labels, screen readers will announce the label text when the user reaches the related control. In many browsers, clicking on the label with the mouse will also move to the control associated with it, providing a greater click area for those with dexterity challenges, and allowing a user who enlarges the screen to find the control that is tied to a label when both are not visible on the screen. Label text should provide any information needed to complete the form field, such as informing the user if the field is required to be completed, the format in which dates should be entered, or password length and character requirements. One reason why form field requirements must be included as part of the label text is that many screen readers will switch into a “Forms mode” that passes the keystrokes onto the web page while a form is being completed, preventing the user from reviewing surrounding text while moving between form fields. Including form field requirements as part of the label text will ensure that these requirements are not missed by the user. Label text for a form control should also be unique to the other labels on the page. Using the same label text for multiple controls on a page will make it difficult for users to move to and identify the desired form control. In addition, related form controls should be grouped together. For example, if a form requests both the billing address and shipping address of a submitter, many of these form field labels will be the same for both sections of the form. Grouping all of the “billing address” fields into one group and all of the “shipping address” fields into another group will allow users to distinguish between these sections of the form. Similarly, multiple form controls which are part of the same question should be grouped together. This would include a question for which radio buttons to answer “yes” or “no” are provided, or any question for which multiple answers have been provided. If controls cannot be grouped together, then the name of the group should be included at the beginning of the label text of each control in the group. For example, “Billing Street Address” and “Billing City” will avoid confusion with the shipping address form fields.

Users must be able to easily locate and recover from errors. This includes placing error messages in a consistent location that users can discover easily. If the error message appears after the user submits the form, the message should be placed at the beginning of the form. Error messages which appear after or below the form may be confusing or difficult to locate by users of assistive technologies. The message should allow the user to locate the field containing the error and clearly describe the changes that must be made to the field. Other methods which can aid the user in locating and correcting the error include providing in-page links that move the user from the error message to each field needing correction, moving the user’s focus directly to the field needing correction, and updating the label text of each field containing an error to include the error message so it is announced when the user tabs to the field.

Web pages which require the user to respond within a certain amount of time must give the option to extend the session before time runs out. Some users may require additional time to complete the form beyond the amount provided by the website when no activity is detected. Before the user is about to be logged out of their session, around one minute before time runs out, provide a warning that time is about to run out and give the option for the user to continue their session by pressing a keystroke. This is best accomplished using a JavaScript alert dialog as it will appear on top of the form, will be automatically announced by screen readers, and can be dismissed using a keystroke. If the alert is being provided directly on the web page, methods should be used to bring it to the user’s attention such as setting focus to the message and using a WAI-ARIA live region to cause it to be automatically announced by screen readers.

Techniques

Making online forms accessible requires that all form controls are tied to unique, descriptive labels. Techniques for associating labels with form controls using HTML source code can be found in the Electronic Forms section of the “guide to the Standards” document on the U.S. Access Board website. Information about warning users when their session is about to expire can be found in the Time Delays section of the “Guide to the Standards” document on the U.S. Access Board website. Additional information and techniques for making the various aspects of forms and form controls accessible can be found under the Forms SSA Accessibility best practices Library on the U.S. Social Security Administration website. Additional details and techniques can be found in the General Form Accessibility section of the “Creating Accessible Forms” article on the Web Accessibility in Mind (WebAIM) website. Techniques for making Adobe Acrobat PDF forms accessible can be found in the Designing Accessible Forms section of the “Creating Accessible PDFs Tutorial” on the U.S. Department of Veterans Affairs website. Techniques for making Adobe Flash form controls accessible can be found on the Providing Accessible User Interface Controls section of the “Creating Accessible Flash Course” on the U.S. Department of Veterans Affairs website.

How to Test

Forms can be tested using the keyboard to verify that they are keyboard-accessible. Testing form keyboard accessibility requires moving through the form using only the tab key to confirm a logical tab order, operating all form controls using the keyboard alone, and submitting the form to confirm completion and proper setting of keyboard focus. In many browsers, you can confirm that form controls are correctly tied to labels by clicking on the label which will cause the form control to become active.

Many automated tools can scan the source code for missing labels and other indicators that impact accessibility. A good way to verify that forms are accessible is to test them using a screen reader. Some screen readers include commands which will bring up all form controls into a list. This not only allows one to easily move to and locate form controls, but it also allows the tester to visually confirm what the screen reader will announce as the label for each control. Many screen readers, in an effort to improve access to forms which are not coded correctly, will try to “guess” what text should be used to label a control when no label is directly tied to it. The screen reader will look for text near the control and use this as the control’s label. In some cases this “guess” may be correct, and for others it may not provide any meaningful information. Most screen readers do not have the option to disable this feature. This means that while using screen readers to test electronic forms is a great way to confirm that they can be completed by users who are blind, it does not guarantee that they are coded properly. Currently the NVDA screen reader does not use this “guessing” technology, so it can be used as an effective tool to test whether forms have been correctly labeled. For information on how to use JAWS for Windows to fill out and submit online forms, refer to the Forms section of the “Surf’s Up!” tutorial on the Freedom Scientific website. Another tutorial which explains how to use screen readers such as JAWS to fill out online forms is the Forms section of the Adaptive Services Internet Classroom tutorial from the Mart Luther King D.C. Public Library.

The only way to be certain that forms have been properly coded is to manually review the code. One effective strategy to reduce the amount of time required to complete this task is to use an automatic tool to scan the web page for potential errors, and then manually review any areas of the page which the tool notes as questionable.

# Links

Users with disabilities use the displayed text of a link to determine what will occur if the link is followed. This is true when the user has access to the text surrounding the link and is reviewing the link within context, or if the user is moving between links and is reviewing the link text out of context from the surrounding text. If link text is not provided, vague, or does not describe the location where the user will be taken by the link or the action that will be performed by the link, users will be uncertain whether choosing the link is desirable. Likewise, if the same link text is used for multiple links on the page which go to different locations or perform different actions, the user may be unsure which link to follow. This is especially true where the text of “click here” or “read more” is used for many links on the page with different locations which is currently prohibited by OCTO standards. Imagine a user of speech recognition software who is trying to choose a link by dictating the link text of “click here,” or a user of a screen reader who is reviewing a list of links on the page or moving between all the links on the page. Vague link text such as “click here” which does not describe the page that the link will take the user to provides no guidance for the user to choose the correct link.

Too many links on a page can impair movement for keyboard users and users of screen readers. When links are provided on every page to jump to different sections of the website, users will have to read or move past these links every time if a method is not provided to skip over them. This can make browsing slow for a user who moves to links using the tab key, or for the user of a screen reader who hears the page read in order from beginning to end.

Requirements

All links on the page must provide clear text which describes the purpose of the link. For links which take the user to a different location, the link should identify the title of the location. For example, the links on the DC.Gov home page of “Ask the Mayor,” “Subscribe to Emails” and “Agency Directory” inform the user that a page will be loaded which allows the user to access these features when the respective link is followed. If a link performs an action such as printing a page or deleting an entry, the link text should indicate the action that the link will perform. If following a link will open the page in a separate window, this should be included as part of the link text, such as “Privacy Policy (opens in new window).” Whenever possible, the user should be able to determine the purpose of the link without needing to review the text around it. This is because the user may review a page by tabbing through all elements on the page, and because many screen readers will present all links on the page in a list, allowing the user to quickly review, locate and choose the desired link. Users will also be unable to review text surrounding a link if their screen reader is in a “Forms” mode which passes keystrokes directly onto the web page, disabling the ability for the user to review text which cannot be reached using the tab key. The link text should allow the user to be able to distinguish the link from other links on the page. This means that the link text should be unique to the page. Multiple links containing text such as “click here,” “edit,” or “delete” but which go to different locations or perform different actions will make it difficult to pick the intended link. The link should include the target, object or destination as part of the link text, such as “delete message with subject ‘upcoming events’” or “more from article ‘Mayor Opposes Tax Cuts’.”

Links should be provided to skip over items which are repeated across web pages. For example, many websites will have the same links at the beginning of each page, allowing the user to jump to different sections of the website. Keyboard users will have to tab past these links on every page, and users of screen readers will hear these links being announced every time a new page loads. A method should be provided which allows users to skip past links or information which is repeated on every page so they do not have to encounter it every time.

If the link leads to a file other than a web page, the link should state the file type and relative size so that users can decide if they want to follow it. For example, a link to an annual report posted as an Adobe PDF file should use link text similar to “2012 Annual Report (PDF, 64k).” This will inform the user that they will be downloading a file and alert them to the fact that they will need to have the proper application installed to read the document. Be advised that the current Section 508 standards require that for files which are not web pages, only accessible files and applications may be posted to websites, and that a link must be provided on the page to the accessible version of the application needed to read the file. Links should never lead to files that are not directly accessible. For example, a link should never directly lead to an image, as no alternative text can be provided within the image file. A better practice is for the link to lead to a web page that offers alternate versions of the file. For example, instead of having a link pointing directly to an image file, have the link point to a web page which contains both the image and any text descriptions which are provided either directly on the page or linked to in a separate file.

Links should respond to keyboard actions. Users who are blind and who have limited mobility may use the keyboard to browse web pages, and providing links which only respond to the mouse may cause challenges for them to use the website. Generally links that do not respond to the keyboard are created using web scripting languages such as JavaScript. To ensure that links can be accessed using the keyboard, the link should lead to a valid address on the web server or a location within the web page.

Techniques

While Section 508 does not have any direct requirements for links, the Web Content Accessibility Guidelines 2.0 does have requirements for links both at the minimum (level A) and maximum (level AAA) conformance levels. Additionally, many of the other Section 508 standards such as providing sufficient information about user interface elements, are relevant to links, and the Section 508 standards will likely include the WCAG 2.0 requirements for links when they are updated. Techniques for providing meaningful link text can be found in the How to Meet Success Criterion 2.4.4 and How to Meet Success criterion 2.4.8 on the W3C-WAI WCAG 2.0 website. For general best practices about link text, refer to the blog entries On the Accessibility of Links and Methods for Indicating the Purpose of a Link. Additional best practices for links can be found in the Links section of the “Accessibility Best Practices Library” on the U.S. Social Security Administration website. Techniques for creating accessible links in Adobe Acrobat PDF documents can be found in the Creating Accessible Links section of the “Creating Accessible PDFs Tutorial” on the U.S. Department of Veterans Affairs website.

For websites which contain the same links on every page to allow the user to jump to different sections of the website, a link should be provided at the beginning of the page which moves the user to a point on the same page past the repeated information. For more information on allowing users to skip past repeated links and information, refer to the Navigation Links section of the “Guide to the Standards” on the U.S. Access Board website. Techniques for allowing users to skip past repeated links and information can be found on the How to Meet Success Criterion 2.4.1 on the W3C-WAI WCAG 2.0 website.

How to Test

Links should be tested from the keyboard to confirm that they can be reached using the tab key and that they can be followed using the keyboard. They should also be tested using the keyboard with assistive technology running at the same time in order to confirm that no issues prevent assistive technologies from accessing the links. Several screen readers include a feature to display all of the links on the page in a single list. This not only assists in quickly reviewing and selecting links on a page, but it also allows the tester to visually confirm the text that will be announced for a link by the screen reader. This is especially helpful when additional text is provided for a link to be announced by the screen reader to aid with understanding the purpose of the link that is intentionally not shown on the screen. For information on how to use JAWS for Windows to access links, refer to the Navigating Web Pages section of the “Surf’s Up!” tutorial on the Freedom Scientific website.

If a link or other method is provided to skip past information which is repeated on every web page, this method should be tested from the keyboard. The link should be the first link on the page which the user can reach with the tab key past any address bars or toolbars included with the browser. When the link has been reached, it should be followed using the keyboard and the page should scroll down to the beginning of the content which only appears on the current page. If a screen reader is running, the screen reader should begin reading at this point after the “skip navigation” link is accessed. Pressing the tab key from the point where the “skip navigation” link moves the user should resume past the repeated links and into the new content.

# Frames

Frames are used for presenting multiple pages on certain areas of the current page. This is done through the use of multiple, independently controllable sections on a Web presentation. This effect is achieved by building each section as a separate HTML file and having one "master" HTML file identify all of the sections. An IFrame (Inline Frame) is an HTML document embedded inside another HTML document on a website. The IFrame HTML element is often used to insert content from another source, such as an advertisement, into a Web page. Although an IFrame behaves like an inline image, it can be configured with its own scrollbar independent of the surrounding page's scrollbar. A Web designer can change an IFrame's content without requiring the user to reload the surrounding page. This capacity is enabled through JavaScript or the target attribute of an HTML anchor. Web designers use IFrames to embed interactive applications in Web pages, including those that employ Ajax (Asynchronous JavaScript and XML), like Google Maps or ecommerce applications.

Requirements

All frames or iframes on the page must have a descriptive frame title and name. A descriptive title gives the user information that will assist them in the determination of the frame content. Users are able to navigate to specific frames on a page, and by providing an appropriate title, the user will be able to accurately choose the frame they require in order to continue with their navigation. Also, many screen readers have the capability to navigate among the frames using the CTRL+TAB and CTRL+SHFT+TAB keystroke combinations. This will allow the users to move to and from frames on a page, given that they are structured and titled properly.

When frames or iframes are used on the page, the frame size must be defined with a relative unit of measurement (i.e., percentages or em). Using a relative unit on the frame or iframe allows for the elements to be adjusted automatically to flow the page content within these containers nicely (i.e., without overflowing the page content) when the browsers are needed to be re-sized.

Techniques

Frames can be made accessible by providing a meaningful frame title and a name for each frame on the page. For information on making frames accessible in web pages, refer to the Frame section of the “Guide to the Standards” on the U.S. Access Board website.

Additional information can be obtained by reviewing the Creating Accessible Frames article on the Web Accessibility in Mind (WebAIM) website.

How to Test

Frames and their appearance and structure on a page can be tested using screen reading software while tabbing through the frames. When a frame is in focus, the screen readers will be able to read the name and the title on the frame. For more robust screen readers such as Freedom Scientific’s JAWS software, the shortcut key Ins+F9 can be used to bring up the frames list, allowing for the tester to ensure that all frames are listed, and that they include an appropriate title.

Frames can also be tested manually. Manually inspecting the source code of the page or document can identify frames used, and allows for the tester to ensure that the appropriate information is being used.

# Page Structure

Appropriate page structure, defined as properly ordered and tagged HTML source code or properly tagged content in a PDF is vital when delivering accessible web sites, applications and documents. Screen readers rely on properly marked up source code, whether it be HTML or the tag structure of a PDF, to render to the user the appropriate information. Because non-sighted users do not have the benefit of the visual representation of a page or document’s content, they must rely on the page structure in order to view things like proper reading order of information, or informative elements such as the page title, headings and list items, as well as to render the proper reading order within the document.

Requirements

Many issues should be addressed when it comes to Page structure and Text Presentation. Because non-sighted users lack the benefit of the visual representation of a page, developers must ensure that the reading order of a page or a document follows a natural flow, and matches that of the page or document’s contents. Normally, assistive technology, much like sighted users often do, reads the page content in the top-to-bottom and left-to-right reading order.

Users of screen readers cannot see the visual structure of a page. Therefore the webpage should have proper and semantic markup that the screen readers are able to understand and read. For example, a visual heading on the page cannot be utilized as a heading until the heading is properly structured. This includes but is not limited to ensuring the appropriate tags are used, that they are placed in the correct order, and that they are not used unnecessarily. Proper heading structure on a page or document is vital in ensuring that content is organized in a sensible order, and allows users of assistive technologies to access it with ease. Proper page structure and heading structure also assists users of screen readers with page navigation, as they are able to access specific areas of content based on the headings, through shortcut keys and the heading list found in more robust screen reader software such as JAWS.

It is also important that the title of the page is meaningful within a website because the page title informs the users about the nature of the content that the page contains. Without proper page titles, a user who visits a specific page in a site or a certain document cannot be sure they have arrived on the correct location without having to investigate the page’s content.

Also, the text on the page should be resizable. That is, the user of the page, especially a low vision user who might rely on screen magnification software, should be able to increase or decrease the text on the page from the browsers’ text re-sizing tools. Avoiding the use of absolute sizing is necessary in order to allow for this text adjustment to occur.

Techniques

Informative and technical information can be obtained regarding the page structure and headings by reviewing the Creating Semantic Structure article on the Web Accessibility in Mind (WebAIM) website

Techniques for making accessible headings for Adobe Acrobat PDF document can be found in the Creating Accessible PDFs with Adobe Acrobat Professional tutorial section on the U.S. Department of Veterans Affairs website.

How to Test

A good way to verify that the headings are accessible is to test for them using a screen reader. When the tester navigates to a heading on the page or in a document, they are able to hear the heading information read back to them. Screen readers with more advanced options allow for the user to pull up a heading list, and to navigate from one to another. This allows users of screen readers to easily locate and move to headings on the page or in the document.

Another way to verify that the headings are present is to manually inspect the source code of the page. Testers should be looking for proper <h> tags in the HTML source code. When working with documents, such as a PDF, the tester is able to manually inspect the tag tree, to ensure that there are headings in place and that they are being used in the correct order. And if working with word documents, the tester can go to View  Check the Navigation Pane checkbox to view the heading structure of the document. Through this, the tester can verify headings are being used properly.

Testers can verify that page titles are being used and used correctly by viewing the top of the browser bar. Viewing the information displayed here will tell the user that a. a title is being used and b. what the title is. It is from this display that the tester can determine if the title is being used correctly.

# Lists

Lists are an important method for grouping related elements together. Users of screen readers depend on properly structured lists in order to gather related information, and when proper list structure is not used, and instead visual elements are put in place to give the appearance of a list, users of assistive technologies will not be able to distinguish between the related items. For example, when developers use graphic elements such as images of bullets next to related text, visually, related items will be grouped together, but a screen reader will not render that relationship to the user. Also, using this method will not notify the user of the beginning and end of the list or the number of related items in the list. Using list item structure enclosed in proper ordered or unordered list containers will ensure that the appropriate relationship is conveyed to the user.

Requirements

All related content on a page presented visually as a list must be structured as a list, in a page’s source code or a document’s page structure. This includes the use of ordered or unordered lists and list item elements. The dependency on graphic elements to distinguish list items will not present the hierarchal relationship to the user. When nested lists are used, e.g. parent (or top level) lists that contain child (or level two) sub-lists, developers and content editors must ensure that that relationship is conveyed through proper structure. This would include using standard list item elements as well as making sure that they all fall within the same list container.

Techniques

Using appropriate list item elements and placing them within list containers in the proper order is vital to ensuring that hierarchal relationships of content are exposed to the user in the correct manner. Informative and technical information can be obtained regarding the lists structure by reviewing the Creating Semantic Structure article on the Web Accessibility in Mind (WebAIM) website.

When creating lists in Microsoft Office products, developers and content editors should always utilize the insert list feature. This will ensure that when the document is converted to other content delivery methods (for example PDF), these list structures convey. For more information on creating lists in Microsoft Word, visit the Microsoft website (http://office.microsoft.com/en-us/word-help/ins-and-outs-of-bullets-and-numbering-in-word-HA001137679.aspx)

When creating list structure in HTML, proper <ul> and <ol> tags (for unordered and ordered list containers respectively) and <li> (list item) tags must be used. For more information on creating ordered and unordered lists in HTML, please visit the W3 Schools website.

How to Test

A good way to verify that the lists are properly structured and accessible is to test them using a screen reader. When testing with a screen reader, a user can navigate to the list in question, and the screen reader will announce the beginning of the list and the number of list items contained in that list. Then, upon navigating through the list, when the tester reaches the end of the list, the screen reader will announce the end. This will validate that the list is at a minimum, enclosed within appropriate list containers, and depending on the number of visual list items present on the page, how many have been properly structured as list items. Testers may also manually inspect the source code of the list structure on the page or document, to ensure that list containers are present and the items within them are placed in appropriate list item containers.

# Style Sheets and Styles

When developers wish to change the way information is displayed on a page, they can turn to style sheets. Style sheets can be used to present text in different colors and fonts, as well as place content in specific locations on a page. This can include blocks of text in a specific position on the page, as well as displaying elements such as background images. When working within documents (such as Microsoft Office documents), content editors can utilize built in styles in order to control how their documents are viewed by the end user. When used appropriately, these styles will convert over when exporting to document types such as PDF.

There are several issues that can arise when using style sheets on a page. First, when pages are dependent on style sheets to expose certain content. When CSS background images are used for vital elements, and colors are removed from the page, those elements will not appear. Without proper textual alternatives (inline text to coincide with the background images), low vision users who disable colors on a page will not be able to see these elements. Also, when relative units such as percentage are used for sizing the text, users of disabilities such as low vision and color-blind users are able to increase (or decrease) the text through the browser, or through the document rendering software. However, when pixels and points are used, this can prevent users from adjusting the size of the content to their own preferences. In documents, when the style settings are not created at the source document level, they will not transfer over to exported files. This can cause issues down the road as documents such as PDFs are created, that will end up lacking the appropriate styles.

Requirements

Users must be able read web pages correctly when style sheets (or at a more basic level color) are removed. This includes reading order being correct, all active elements on a page being visible with at least a textual alternative. Users who choose to must also be able to utilize their own external style sheet from within the browser’s built in accessibility menu in order to override the default style sheets. Pages must allow for this user settings, and cannot prevent the user from choosing their own style settings. This ability greatly helps the users for reading the page content according to their need.

Users must also be able to resize the page content or text on the browser. Developers must be certain to avoid using absolute sizing when using styles on a page. Absolute sizing will prevent the user from being able to adjust the sizing on the page. Instead, developers must use relative sizing, as this will base the text and content size on the percentage of screen magnification.

Techniques

Developers must ensure certain rules are followed in order to make documents that utilize style or pages that utilize style sheets accessible. When creating documents, content developers must utilize the built in styles menu in products like Microsoft Office. These styles can be adjusted to meet organization styles guidelines and branding styles. By using the style menu, which can be found under the Home tab of Microsoft Office, this will ensure that when the document is converted over to secondary software such as Acrobat PDF. When creating web pages, developers must ensure that pages can be viewed correctly when style sheets are removed and avoid using absolute sizing as this will prevent users from adjusting the content on a page through magnification.

Informative and technical information can be obtained regarding the style sheets by reviewing the Styles for Checking Color Reliance article on the Web Accessibility in Mind website.

For information about the reading order on the page refer to the Readability section of the “Guide to the Standards” on the U.S. Access Board website.

Techniques for more information about the hidden page contents on the page can be found in the article “CSS in Action: Invisible Content Just for Screen Reader Users” on the Web Accessibility in Mind website.

How to Test

The best way to test that style sheets are being used in an accessible manner is by disabling and re-enabling the attached style sheets on the browsers’ View Menu (in I.E., View – Style sheet – No Style (or Default Style)). By disabling and then re-enabling the styles, the tester should be able to confirm the logical reading order is correctly in place, as when style sheets are removed, the page reverts to the reading order based on the current source code. Also, when style sheets are removed, the tester can verify that all content is in fact displayed on the page.

The tester can also import a customized external style sheet via the browser’s Accessibility menu (in I.E., Tools – Internet Options – Accessibility) to test that the attached (as well as the browsers default) style sheets are overridden with the customized style sheet.

The tester should also ignore colors through the browsers’ Accessibility tool (in I.E., Tools – Internet Option – Accessibility – color checkbox). This will allow the tester to verify that colors (or background images) are not being used for vital elements on the page, as these elements will disappear when colors are ignored.

The tester can test whether the page content and text is resizable, This can be accomplished from the browsers’ View menu (i.e., View – Text size - larger)

# Embedded Content

When developers and content editors utilize embedded media content in a document or on a web page, there are several things to keep in mind and take into consideration in order for this content to be used in an accessible manner. Some users may not have the appropriate software installed in order to view this content. For example, Flash content requires that a Flash player be installed. While it is safe to assume that many users may have this plug-in, there still must be a mechanism in place in order to download it for those that don’t have it installed. Providing a direct link to download the plugins will allow users to install the required software, then access the embedded content. Embedded content must also be directly accessible. Setting the Flash content’s wmode to window will allow assistive technologies to access the content and its controls. Textual equivalents are also important for live audio or video only embedded content. This can be found in the form of audio descriptions for video only content, and captions or text equivalents for audio only content. Without these conditions and equivalents in place, users of assistive technologies will have a difficult time accessing the content.

Requirements

Users must be able to open the embedded content on the page. Developers and content editors must never assume that all users accessing their pages and documents will have the necessary plug-ins installed in order to view embedded Flash, PDF or other media type content. When embedded content such as a Flash video or a PDF document are present on the page, a direct download link must be present on the page or in the location where the document is accessed from, to download the required programs or plug-in. This helps users with disabilities to download the plug-in right way without leaving the current page or location where they open the embedded content

Users with disabilities must also be able to directly access embedded content on the page. Flash content must be set up to expose the accessible controls. For example, a keyboard user must be able to use the Tab key to access the buttons used to play or pause a Flash video, and those controls must expose accessible properties to the user. PDF content must be tagged in a way that users of assistive technology can read the PDF’s content. Also, a text description of the functionality and, when possible, a textual alternative of the embedded content should be present on the webpage or the document. The embedded content must be accessible before these are embedded on the webpage or on any other media. For example, Flash video can be made accessible in the Flash authoring environment before the video can be embedded on the webpage. If it cannot be made accessible, then an accessible equivalent must be made available.

When live audio-only and video-only presentations are present on the page, an alternative text description of the presentations must also be present on the page so the users of disabilities are able to read the presentations. For video-only presentation, the captioning of the video must be available in an accessible manner so that the screen reader users can use the caption to listen to the text-equivalent of the video presentation. Similarly, a text description of the audio-only presentation must be available on the page in an audio transcript format.

Techniques

Information regarding the accessible plug-ins for embedded media content can be obtained in the Applets and Plug-Ins section of the “Guide to the Standards” on the U.S. Access Board website.

Information regarding the multimedia presentations and the text alternative to these presentations can be obtained in the Multimedia Presentations section of the “Guide to the Standards” on the U.S. Access Board website.

Techniques for creating an accessible Adobe Acrobat PDF document can be found in the Creating Accessible PDFs with Adobe Acrobat Professional tutorial section on the U.S. Department of Veterans Affairs website.

Techniques for creating accessible Flash content and embedding Flash content on the website can be found on the Creating Accessible Flash Course on the Department of Veterans Affairs Website.

How to Test

The best way to test that a download link is available for the embedded content is to visually scan the page or document in question, and ensure that a link to the appropriate plugin (based on the media found on the page) is available. Also, testers can manually inspect the source code on the page and determine if the link is pointing the user to the correct download page.

The best way to verify that the Flash contents (i.e., a Flash Video) are accessible is to test them with a keyboard and using a screen reader. Navigating with the Tab key, the Flash contents should gain a keyboard focus. Testing with the screen reader, when the tester navigates to the Flash contents with the Tab key, the tester will be able to hear the information of the focused items.

The MSAA Object Inspector tool allows users to inspect the MSAA name, description, role, Kbshortcut, state, and value properties without relying solely on assistive technologies like screen readers. In addition, Object Inspector has a focus rectangle tracker that will help in tracking down visual focus related issues. This tool can be used to determine if the embedded content is accessible (via the focus tracker) and whether the correct (if any) control information is being relayed to the user.

When testing PDF content, the tester may use the keyboard and a screen reader to ensure that they can access the content, and that the correct information is being rendered to the screen reader.

# Animation

When content incorporates the use of animation, there are several accessibility issues that developers should keep in mind. First and foremost is user impact. It is important to remember what the end user might experience should they come across animated content on your site or document. Users with cognitive impairments might become distracted by animated content. For this reason, the ability to pause, stop or play animated content is important. Developers must also provide a non-animated equivalent for users to access the same information provided in the animation. An example of this would be an animated clock. A separate link should be provided that will allow the user to move past the animation and access an accessible equivalent, for example a site that displays the date and time at the top.

Requirements

When animation that provides meaningful content is utilized in a web page, developers must provide the animated content in steps that can be reviewed one at a time. An example of this would be to provide a pause, play, stop and forward/backward control for a slideshow animation displaying top stories, giving the user the ability to stop on a specific story, or move forward and backward through the list.

When animation is present, developers must ensure that an alternative presentation is available without animation, that will allow a user to skip past the embedded animation and access the non-animated equivalent. This is accomplished through the addition of a link that will allow the user to access the non-animated equivalent directly.

Techniques

In order to ensure that animation is used in an accessible manner, developers and content editors must ensure certain conditions are met. For one, they must include a method that allows users to skip past animated content to a non-animated equivalent. This can be done by implementing a link that takes the user to an accessible equivalent. In the example above, if the animation on a page is of a clock that gives the user the current time, developers would be required to provide a link that takes the user to a site like Time.gov to allow the user to check the time in an accessible format. Developers must also ensure that users are able to control the rate at which content is revealed to them, through the use of pause, play and stop controls, as well as controls that allow the user to move forward and backward in the animation sequence. For more information about accessible animation controls, visit the US Department of Veterans Affairs “Creating Accessible Flash Course.”

How to Test

In order to test for accessible animations on a page or in a document, testers may verify that when animation appears on a page or in a document, that controls exist that allow the user to navigate within and control the animation. Using the keyboard, testers must ensure that these controls are keyboard accessible. Testers can also manually inspect the document or page’s source code, in order to determine if these controls exist.

Visual inspection of the document in question can reveal if a link has been provided that allows the user to access an equivalent to the animation that provides accessibility. Ensure that this link can be activated with the keyboard, and that the location it takes the user contains a non-animated equivalent to the animation appearing on the originating page. Manual inspection of the source code can also reveal if this equivalent access link is provided.

# JavaScript

JavaScript is scripting language that assists with creating dynamic content on the web. Without JavaScript, a document on the web will be static and the content on the page will not change until the page refreshes or a user navigates from it by clicking a link or button. JavaScript can be used to validate content on a web page like form field data, insert content on the page dynamically without reloading the page and trigger events or actions to occur by monitoring a user’s actions like mouse clicks and keyboard presses. A JavaScript event is an action that occurs after a user has done something on the page that causes the JavaScript to start.

One powerful feature of JavaScript is the ability the set the focus to a location on the page after an event (after the user does something). The focus may be a blinking cursor on a form field or the dotted rectangle on a link or button.

Requirements

The most important requirement is to ensure that a user can use the page with the keyboard. Often, JavaScript is used after a mouse event has occurred with no consideration for keyboard alternatives.

Next, avoid focus changes that a user may be aware of. This means that the blinking curser (for form fields) or the dotted rectangle (with links and buttons) is not automatically set to a location without the user specifically causing such an action to occur.

When simulated dialog windows are opened, the focus must be placed to the beginning of the dialog window. When the dialog windows are closed, the focus must return to the link or button that caused the dialog to open.

Techniques

For every mouse event there must be an equal keyboard event applied or taken into consideration.

If the mouse hovers over an object and something changes automatically on the page, this must also occur when a user navigates to the object with the keyboard.

When clicking an object with the mouse causes something to change, the change or event must occur when the keyboard is used.

A button must be present (next to) a combo box that explicitly selects a highlighted option in the combo box.

When a page is loaded, it is acceptable to set the focus (the blinking cursor) on a particular form field or section of the page. If content changes automatically below the section where the cursor or dotted focus rectangle is on the page, the focus does not have to be set by the developer. If content changes above the user’s current location on the page, developers may set the focus to that location. It is not acceptable to set the focus anywhere on the page unless an object like a link or button has been activated or selected with the Enter key or Spacebar key on the keyboard.

How to Test

The keyboard must be used to test for the accessible use of JavaScript.

The Tab key is used to navigate through web content in a logical manner. The Tab key will set focus to form fields, links and buttons on the page. The keyboard combination Shift+Tab is used to move the focus backwards on a page. At any time, the user must not get stuck on an object when Tab or Shift+Tab is pressed. Additionally, the focus must not be placed elsewhere unless the spacebar or Enter key is pressed on a button or link by the user.

When a radio button group is navigated to, each radio button within the group must be accessible with the arrow keys. To navigate beyond a radio button group with the keyboard, Tab or Shift+Tab will achieve this.

A keyboard user must be able to inspect combo box options with the arrow keys and open a combo box with Alt+Down Arrow without the page refreshing or change of focus. A button must be present next to a combo box that can be accessible with the Tab key and activated with Enter or Spacebar.

# WAI-ARIA

WAI-ARIA (“ARIA”) is a technical specification by the W3C. This specification is similar to HTML in structure but is offered to increase the accessibility of standard HTML and JavaScript implementations for assistive technologies; specifically screen reading devices for users that are blind. The function of ARIA is to help users without vision to gather information such as the name, state, value and role of an object.

To help explain name, state, value, and object better and give an example of the purpose of ARIA, we will take a progress bar into consideration. In HTML, there is no progress bar object available for developers but it does not keep developers from creating one with the HTML objects that are available. Visually, one can see the object and verify that it is a progress bar. However, someone who cannot see will not be able to discern or understand this information. Therefore, ARIA can be used to identify the “role” of “progress bar”. Next, the “name” can be identified by providing the object a label such as “loading”. The current value can be provided by setting the appropriate ARIA values, often in a percentage such as 80%.

As the internet is becoming more rich or interactive, many developers create custom form fields like checkboxes to make them more appealing visually. The challenge with ARIA is that it is not supported by some browser and assistive technology combinations. The “state” of a checkbox, for example, can be “checked” or “unchecked” at any time. If a custom checkbox is created, this information must be provided to a user and ARIA can achieve this. However, standard checkbox objects automatically convey the name, state, and role information without any additional ARIA attributes. Therefore, developers should use ARIA to enhance accessibility only when it cannot be achieved with native HTML objects.

Finally, ARIA can be used to define or structure areas of the page that are known to be dynamic. ARIA can be used to define the areas of the page that contain or may contain information that changes automatically by structuring these regions as “landmarks” that allow a user to navigate quickly to it when the user’s technology supports it.

Requirements

Developers must correctly set ARIA properties including roles based on the intended purpose. When ARIA attributes (state, roles, and properties) are inappropriately used, assistive technology may not correctly function as expected. The applicable ARIA roles can be referenced from the WAI-ARIA Roles Model page. The applicable ARIA states and properties can be referenced from the WAI-ARIA Supported States and Properties page.

Developers must not create custom links, buttons, radio buttons, check boxes, combo boxes, multi-select list boxes or edit fields. Developers must use standard HTML objects for these controls because ARIA is not fully supported by all browsers and assistive technology combinations.

Because of the rich internet feel, JavaScript is an integral part of ensuring that the name, state, value and roles defined with ARIA are updated dynamically. Developers must ensure that the JavaScript does not prohibit the equivalent use of the keyboard or create focus issues by a user who relies on the keyboard to navigate to and activate objects on the page.

Techniques

Until ARIA is supported by all mainstream user agents (e.g. browsers and assistive technologies), the Web Accessibility Requirements and techniques in this document must be followed to ensure optimal graceful degradation (ability for content to be accessible without requiring a specific browser or AT combination).

How to Test

The MSAA Object Inspector tool allows users to inspect the MSAA name, description, role, Kbshortcut, state, and value properties without relying solely on assistive technologies like screen readers. In addition, Object Inspector has a focus rectangle tracker that will help in tracking down visual focus related issues. The focus rectangle tracker is helpful to follow the programmatic focus within the application and can simulate the focus tracking behaviour of screen magnification software, when such software is not available to the tester.

# The following MSAA properties can be tested with Object Inspector:

**Property | Definition | Example**

**Name:** The label of the element or interface. The accessible Name of a button is “Go”.

**Value:** The contents of edit fields, combo and list boxes, sliders, tree nodes, etc. The accessible Value of a combo box that lists city is “New York”.

**Role:** The function of an element or interface. The accessible Role of an interface is “button”.

**State**: The status of an element or interface. The accessible State of a radio button is “selected”.

**Description:** The explanation of what the element or interface is. The accessible Description of a data table is “The data table shows how many businesses are downsizing and at what rate”.

**DefAction:** The default action of an interface. The default action of a button is “press”.

**Kbshortcut**: The keyboard combination to activate an interface. The keyboard shortcut for the “File” menu is “Alt+F”.

# References and Tools

Automated Testing Tools and Plug-ins:

• Accessibility Management Platform (AMP) - https://amp.ssbbartgroup.com

• WAVE - http://wave.webaim.org/

• Web Accessibility Toolbar - http://www.visionaustralia.org.au/ais/toolbar/

Color Testing:

• GrayBit Grayscale Conversion Contrast Accessibility Tool - http://graybit.com/main.php

• Contrast Analyzer for Windows and Mac - http://www.paciellogroup.com/resources/contrastAnalyser

• Color Contrast Check Tool - http://snook.ca/technical/colour\_contrast/colour.html

MSAA Testing Tools:

• Inspect - http://msdn.microsoft.com/en-us/library/dd318521%28VS.85%29.aspx

PDF Accessibility Tools:

• Adobe Acrobat Pro - http://www.adobe.com/accessibility/products/acrobat/

• CommonLook PDF - http://www.commonlook.com/CommonLook-PDF

Section 508 Guide and Best Practices:

• United States Access-Board - http://www.access-board.gov/sec508/guide/index.htm

• Web Accessibility - https://www.webaccessibility.com/best\_practices.php

Screen Readers:

• NVDA - http://www.nvda-project.org/

• JAWS - http://www.freedomscientific.com/products/fs/JAWS-product-page.asp

Data Tables:

• Web Experience Toolkit (WET) - http://wet-boew.github.com/wet-boew/demos/tableparser/validator-htmltable.html

# Social Media Accessibility

Overview

The trend of using social networking technology to communicate and share information continues to grow and cross demographic boundaries. A recent D.C. Government Agency audit of social media found that 71 percent of respondents use Facebook, 67 percent of respondents use Twitter, 22 percent of respondents use YouTube, and 11 percent of respondents use Flickr. The audit also found that 69 percent of government agencies use social media for event promotion, 58 percent use social media for constituents services, and 42 percent use social media for advocacy. Many organizations, including government agencies, have found social networking to be an effective avenue for engaging and reaching the public. In the District, this turns out to be no less true. For example, 2,732 comments were received about a single piece of proposed regulation, an all-time District record, and there were 98,732 page views in six months for a blog entry about building codes for basement apartments.

As D.C. government agencies begin to use social media networks for collaboration, sharing and brand building, they will need to ensure that their use of social media provides effective communication to people with disabilities. While the number of people with disabilities who use social media continues to increase along with the general population, many social media platforms present accessibility challenges which prevent users with disabilities from fully using and participating in social networks. Social media accessibility challenges can generally be divided into two categories: accessibility challenges with the interface of the social network itself, and challenges with inaccessible information which is published using social media. To solve the first problem of interfaces not being accessible, many users will use alternate interfaces to access social networks which often take the form of third-party client interfaces, smartphone apps, or versions of the website which are intended for mobile devices. Authors who publish information to social media networks should be aware that their information may not be reviewed using the standard social network interface and take steps to provide information which can be understood regardless of the device or interface being used.

# Social Media Networks

As the popularity of social networking climbs, new networks are continuously being created to attract a loyal user base. Facebook and Twitter are used by most organizations to connect with the public because they enjoy the greatest degree of popularity. Many organizations will also use platforms such as YouTube to host and publish videos. While other social networks can and likely will be used by government agencies, it would be impossible to cover every social network that might be used or created in the future and the methods for using them to provide effective communication to people with disabilities.

Facebook

Facebook is a free, interactive social networking platform which allows users to connect. Facebook connects friends, family members, co-workers, classmates, clients and customers. Users can create a personal profile, add other users as friends, exchange messages, share information and media, comment on published information, join interest groups, and create a profile page for an organization. While Facebook was initially launched to connect students with each other, it is important to note that the 55-64 age group is approaching the 13-17 age group, demonstrating greater penetration within the “baby boomer” generation. Organizations, recognizing the growth of Facebook usage among older adults, have begun using it as a platform for outreach and engagement with customers and clients.

Many people with disabilities use Facebook in order to connect with their friends, family and to access information. The primary Facebook website contains accessibility problems which may change when parts of the interface get redesigned. This usually results in users of assistive technologies not having access to certain features on the site, or requiring them to use workarounds in order to gain access to those features. In 2008, Facebook partnered with the American Foundation for the Blind to begin addressing some of the accessibility issues with the site. While some issues have been resolved, certain features are still not accessible and others may require the user to be using a recent version of a screen reader and web browser to benefit from the provided accessibility support. Facebook offers a help page on Using Facebook with Screen Readers and Other Assistive Technology. Some users of screen readers will use the versions of the Facebook site intended for mobile devices, m.facebook.com. The mobile interface does not contain as many features as the main site, however it is preferred by some users of screen readers because it provides a simpler, more streamlined interface and offers keyboard shortcuts for the main features of the site. Other users may gain access to Facebook through apps on their smartphone. This includes the Facebook-developed app along with third-party apps that can access Facebook. Reviews of the compatibility of social networking apps with the Voiceover screen reader which is built into the iPhone can be found in the AppleVis iOS App Directory. Other users gain access to Facebook through third-party clients such as the Socializer from Serotek Corporation which contain interfaces specifically designed for users of assistive technology.

Twitter

Twitter is a social networking and microblogging service that allows users to send and receive messages which are called tweets. A tweet is a post containing no more than 140 characters of text which is displayed on the profile page of the author, and delivered to the author’s subscribers which are known as followers. Twitter is generally used to provide updates in real time, share information and resources, engage online discussions and promote events and activities. Twitter also provides a search feature which allows people to search for certain words or phrases. To make searching easier, many tweets on the same topic will contain what is known as a hashtag – a common word or abbreviation used to identify the topic which begins with the “#” symbol. For example, many tweets on the topic of accessibility contain the hashtag “#a11y” to represent accessibility, there being 11 letters between the “a” and the “y” in the word accessibility. Other conventions adopted by users of Twitter include forwarding a tweet onto the author’s subscribers by “retweeting” a tweet which involves placing the letters “rt” before the username of the author and the tweet. A Twitter username is represented on Twitter by placing an “@” symbol before the username. This identifies the user or directs the tweet towards him or her, and a tweet containing the “@” symbol followed by a username is known as a “mention.” This is the system used to reply to tweets made by other users.

The text-based nature of Twitter makes it one of the most accessible forms of social networking, however many accessibility issues exist with the interface on the Twitter website, including a confusing layout structure, lack of keyboard access and inability to change text size. For this reason, most users for whom the Twitter website does not work well with their assistive technologies will use alternate interfaces or third-party clients. One web application which was developed to provide alternate access to Twitter through an interface which meets WCAG 2.0 requirements is Easy Chirp. Easy Chirp was designed to provide a more intuitive layout to the web interface, keyboard accessibility, and support for screen readers. While limited on features, some users with disabilities access Twitter through simplified web interfaces developed for mobile devices such as m.twitter.com. Others will use Twitter apps specially designed for their smartphones. In addition, several non-visual Twitter clients have been created by developers who are blind for Windows users of screen readers. Examples of these include The Qube, Klango Twitter and Twitmonger.

YouTube

YouTube allows users to upload, view and share videos. While other video hosting platforms exist such as Metacafe, Vimeo and UStream, ComScore ranked YouTube the dominant provider of online video in the United States in 2010. Alexa ranks YouTube as the third most visited website on the Internet, behind Google and Facebook. In 2006, YouTube was acquired by Google, which resulted in expanded features and better integration with other Google products. Two new features which Google brought to YouTube were video captioning and auto-transcription. These features allowed video creators to add closed captions to their videos using a combination of experimental speech recognition, automatic timing, and subtitling technologies.

The addition of captioning features is one example of how accessibility of YouTube has improved since it was purchased by Google. Accessibility challenges with the YouTube interface are still faced today by users with disabilities, primarily because Adobe Flash technology is used for the YouTube player on desktop platforms. This presents challenges operating the player due to keyboard accessibility issues and compatibility with screen readers on platforms where Flash is not supported by screen readers such as Mac OS. Those who find challenges using the primary YouTube site may use alternate interfaces to the standard player created using JavaScript or HTML5. These include Accessible YouTube, Easy YouTube and Accessible Interface to YouTube. Each player varies in the amount of accessibility support provided including keyboard access, support for screen readers, and display of any provided captions. Users of smartphones and devices based on similar technology such as the iPad, Apple TV and Google TV can often successfully access YouTube using interfaces provided by apps installed on their device.

User Impact

Users with disabilities can experience accessibility challenges with any aspect of social networking features. Most of the challenges will occur while trying to use the standard interface provided by the social networking platform, for example www.facebook.com, www.twitter.com and www.youtube.com. For a review of standard social media interfaces for the most popular platforms as of 2011, refer to Social Media Accessibility: Where are We Today by Denis Boudreau. Several social networking sites use a CAPTCHA during the sign-up process in an attempt to fight spam and fake accounts. CAPTCHA is a method that involves presenting the user with a challenge which, if solved, is supposed to demonstrate that the challenge was solved by a human rather than a machine. Usually this challenge involves recognizing text which is included as part of an image and typing the displayed text into an input field. Screen readers will be unable to read the text displayed in the image, and users with limited vision may have problems reading the text within the image. Some sites do offer alternatives to the standard visual CAPTCHA, such as requiring the user to enter a sequence of characters that is provided using speech. This solution, however, does not work for an individual who is deaf and also unable to solve the visual challenge, such as in the case of a user who is deaf-blind.

Keyboard access will be found to be a problem in situations where the user is forced to use the standard interface. For example, if a user receives a link to a post on a social media site and wants to contribute to the discussion by adding a comment, they may find it difficult or impossible to post a comment using the web interface. Some blogs and forums require the user to solve a CAPTCHA in order to post a comment, effectively shutting out participation by part of the public. Another example of a feature which experiences issues being access from the keyboard is tagging people in a Facebook photo or post. Screen reader users, if asked, would find it difficult to tag people because keyboard access is not provided to allow users to easily tag people in a Facebook photo or post.

When alternatives are not provided for images, audio and video, users may not be able to understand the content in a post. Section 508 requires that alternatives be provided for images, audio files, video files and multimedia (audio that is synchronized with video). Unfortunately not all platforms allow authors to provide alternatives for their media. For example, while YouTube allows authors to submit a caption file for their videos, Facebook does not support closed captioning on their platform. This leaves the author to find other solutions for providing these alternatives. Even when the platform does allow the author to provide alternatives, the alternative must still present the same level of understanding to the user as the regular content. For example, the YouTube auto-transcription service makes mistakes recognizing speech in a video, and these errors must be corrected before the transcript or caption file is published.

Another area which can affect accessibility outside of social media websites is the use of widgets or similar components which are used to display content from social media services on another website, or to post content from a website to a social media service. This is commonly seen in the form of “share” buttons, which allow users to post a link to the page on a variety of social media networks. Unfortunately many of the third-party sharing widgets suffer from lack of keyboard access, missing text alternatives for images, missing titles for frames and inline frames used to embed content, and poor color contrast. This makes it difficult for users to benefit from these services and also creates additional accessibility barriers on the web pages where the widgets are contained.

Requirements

Whenever possible, avoid requiring users to use websites outside of the DC.Gov portal to review or respond to your message. The accessibility of these websites cannot be controlled by the D.C. government, leaving the effectiveness of your communication at the whim of the site owner. As stated before, many users with disabilities are effectively using social networking sites, and others are accessing these same networks using alternate interfaces. While social media is a great tool for reaching and engaging the public, there will be those who find it anywhere from difficult to impossible to access a particular social network. For communication to be truly effective, always have alternatives to social networking platforms that still permit the user to review a message and participate in the conversation.

Images which contribute meaning to the understanding of a post will need to be described. Some social networks allow authors to provide text descriptions for images in particular features of the platform. Facebook allows users to provide a caption for photos uploaded to a user’s photo album. The caption then becomes the link text for each photo in the album, causing it to be announced by screen readers as each photo is encountered. Descriptive captions should be provided for each photo uploaded to Facebook. For images which are posted directly to a user’s wall and come from an outside source, a text description of the image will need to be written as part of the author’s comment in the post, or a link to a description of the image on another page should be included as part of the post or in the comments. Images are posted to Twitter by tweeting a link to the image, as Twitter is a text medium. This means that unless the image is described as part of the tweet, the link should never go directly to the image. Doing so would not make it possible to provide a text description for the image. Instead, post a link which takes the user to a web page allowing the user to access both the image and the text description of the image.

Posts which contain only audio or only video must be provided with a text transcript containing the content of the audio or video. This transcript must include the information needed to gain the same level of understanding that would be gained by a user reviewing the audio or video file. Currently most social networks other than YouTube do not support the uploading of transcript files along with media. In these situations, the transcript will have to be hosted on a separate website, and a link to the transcript would then be provided in the post on the social network. For both audio and video that is synchronized (scheduled to keep the audio and video together by making them appear at certain times), open or closed captions and descriptive video must be available in cases where the audio or the video are crucial to understanding the presentation. Open and closed captions are text versions of the spoken word. They allow users to read what others are hearing via sound. Open captions are part of the video program and cannot be turned off. Closed captions do not appear until the user turns them on and can be disabled at will. For more information about captions and providing captions for video, refer to the Video Captioning on the Web slides by Russell Heimlich and the Making videos more accessible to the deaf and hard-of-hearing slides by Olivier Nourry.

The YouTube service offers several features which assist with providing closed captions for uploaded videos. These include a feature which recognizes speech within the video to generate a text transcript, a feature which synchronizes the captions with the audio, and the ability to create and edit the caption file. For instructions on captioning videos uploaded to YouTube and making the player controls accessible, refer to the Captioning YouTube Video and Providing Accessible Controls on the Ohio State University Web Accessibility Center website. For social media sites which do not include captioning as part of their video services such as Facebook and Ustream, a separate captioned video will have to be provided to make the content accessible to people with disabilities. The captioned video should be hosted on a site which supports closed captioning (or open captioning if closed captions are not supported or desired), and a link provided to the captioned video using the social networking platform. Captions can be provided for videos by using tools such as the Amara project.

Audio description, also known as video description or descriptive video, is a verbal description of the key visual elements of a program which is inserted during natural pauses in the program’s dialogue. Key visual elements are those that would be missed by viewers with low vision and include actions, costumes, gestures, scene changes, facial expressions and onscreen text. Audio description can be found on TV, movies, live theatre, museums and other attractions and on the web. Both the Section 508 and WCAG 2.0 standards contain requirements to provide video description of key visual elements when they are needed to understand a video. In situations where the natural pauses built into the program don’t provide enough time to describe critical aspects of the provided visual information, extended video description can be used instead. Extended video description pauses the video or allows the user to pause the video in order to review a verbal description track which is longer than the length of silence provided in the original video.

Providing audio description requires that a separate audio track be created containing the verbal descriptions. This track can either play alongside the audio track of the video, or the two tracks can be mixed together and the combined track can be played in place of the audio track on the video. Unfortunately as of this writing there are no known social media platforms which support the inclusion of an audio description track along with a video uploaded to the site. This means that the audio description will need to be provided separately, and a link to the track included with the video. Techniques for preparing to create descriptions, determining what information to describe and how to describe it can be found on the Description Key website which is part of the Described and Captioned Media Program. For a list of providers which will produce audio descriptions, see the Description Service Vendors (PDF) document on the Described and Captioned Media Programs website.

When content from a social media or third-party sharing site is included on a web page within the DC.Gov portal, steps should be taken to make sure it is provided in an accessible fashion. Often web pages will choose to display content on their website which originates from a social media site, for example, an agency may choose to display the most recent tweets from their Twitter account. This syndicated content is often embedded inside a frame, an inline frame or similar container. Social media sites make it easy for page developers to display their content on the developer’s web page by providing them with code that can be inserted into the web page. This is also true of social sharing services which create widgets that provide users with the ability to share the page with various social networks. Social media sites also offer different widgets in order to include features of the site on a web page, such as the ability to log in using an account from a social media service, read or post comments using the site’s commenting features, or access other features offered by the site. Often a frame or inline frame may be included on the site to allow for communication between the web page and the social media site which does not contain any actual content, but since the frame or inline frame is not embedded in a manner that allows assistive technology to ignore it, the frame will be visible to assistive technologies which may then provide non-meaningful or confusing information about the frame. Many social media sites and social sharing services generate code which is missing titles for frames and iframes, alternative text for images, keyboard access for controls, and acceptable color contrast levels. There are very few social media sharing services which take steps to make their sharing widgets accessible. When inaccessible code is provided by a sharing service, steps must be taken to detect the accessibility problems and repair the code before the web page is published. These steps can be avoided or reduced if the social media site or sharing service generates code containing fewer accessibility issues. One sharing service which has recently taken steps to make their widgets more compliant with Section 508 is the AddThis platform from Clearspring Technologies. To read about the steps that were taken to improve the accessibility of the AddThis platform, refer to the Making Nine Million Site More Accessible slides by Aaron Jorbin of Clearspring Technologies. Page developers should also verify that any content they display on their web page from a social media site is embedded in the page in a manner that exposes the embedded information to assistive technologies.

# Reaching Constituents with Disabilities

When the above principles are followed to make websites and social media accessible, communication can be provided with people who have disabilities which is nearly as effective as the communication that is provided to others. With a better understanding of how people with disabilities may use DC.Gov websites and social media channels to interact with District agencies, processes can be adjusted and streamlined to improve the interaction experience for all involved.

# Using Websites for Outreach

As mentioned in the Accessibility Overview section of this document, many users with disabilities will access web pages with assistive technologies using either standard or specialized devices. Some users will use the keyboard only or devices which emulate the keyboard to access web content. Others may only use a pointing device such as a mouse or head pointer. Others may alter the display of the page by using a larger monitor, changing font size, colors or other aspects of the page appearance. For these reasons, pages should be coded according to guidelines which allows the content to be interpreted, accessed and displayed in the manner most suitable for the user.

Each agency website should include an accessibility page on their website. This is similar to the accessibility page on the DC.Gov portal which provides general accessibility information about the portal. Accessibility pages for agency websites should include:

• Information about any special features or measures taken to make the site accessible and how these features can be activated

• Any changes that users may need to make to their configuration or to their assistive technologies to take advantage of special accessibility features or improve access to the site

• Any keyboard shortcuts or other alternate methods of operation that have been provided

• Contact information that can be used to report problems accessing information on the website, request assistance accessing information (such as to request assistance in completing a job application), request information in alternate formats, or contact the agency regarding disability or ADA issues

• Any alternate methods of contact provided by the agency such as TTY numbers

• Any additional services provided to people with disabilities by the agency (such as Braille bills or wheelchair rentals) and how these services can be accessed

# Using Social Media for Outreach

As mentioned earlier, social media can be a powerful tool to reach and engage people with disabilities, but social networks can also create barriers to providing effective communication. Users with disabilities may access social networks using any number of applications or interfaces, oftentimes never using the standard website of the social network. Reliance on features available only through the standard website should be avoided whenever possible. For example, a user may not be able to vote in a poll or participate in a real-time instant messaging chat with the interfaces or applications used to gain access to the social network. When special features are used, alternate methods must be provided which allow users with disabilities to equally participate in the activity and contribute in an equal fashion.

The majority of information should be posted on the agency website in addition to or instead of a social network. Users should be moved away from the interface of the social network as much as possible due to the many accessibility challenges which exist with social networking sites. Social media channels should be used primarily to make users aware of new information and to interact with users, then direct the user to a location for additional information where accessibility of the interface is under better control of the agency. An exception to the rule of not using social media platforms to host agency content would be YouTube, which provides more features to make videos accessible than other video hosting sites. The agency website and the social media post should contain links to alternate formats for the information, for example, a version of a video which includes an audio description track. Having the content which is posted to social media channels available via another method such as on the agency website will ensure the information is also made available to those who may be unable to access the information using social media channels.

Agencies should include contact information in their profile to enable users with disabilities to reach them concerning any issues accessing information or interacting with the agency. Agencies should also post updates to social media channels explaining the services they offer for users with disabilities, the different methods for accessing information and contacting the agency, and information on how to report issues or obtain assistance.

# Using Blogs and Newsletters

Blogs and newsletters provide effective methods for providing accessible information due to the fact that the format and presentation are under greater control of the author. Updates are often easier for users to obtain as well due to the fact that most newsletters will push the update to the user through Email, and blogs can push updates to user through real simple syndication (RSS) feeds. In addition, sharing blogs and newsletters among agencies and disability resources can attract new attention from the disability community.

Blogs and newsletters should use simple formatting structures whenever possible. This includes avoiding complex structures such as tables and text in multiple columns. Users may have difficulty reviewing content which contains complex formatting with certain Email clients for newsletters that are sent by Email.

When features are provided which allow readers to discuss the content, these should be provided in an accessible fashion. One major barrier which prevents users from commenting on blog posts is the implementation of a CAPTCHA. CAPTCHA is intended to keep spam from being posted to blogs and forums, however often it is more successful in preventing users with disabilities from participating in the discussion than it is spammers. A better alternative to CAPTCHA is to use an account registration system which verifies the account holder and requires them to log in before posting.

# Providing Alternate Formats and Methods of Presentation

As discussed earlier, alternate formats and methods of presentation should be available for the benefit of those who may find one method difficult to access. For social media posts, this simply involves providing the content in an additional location such as the agency website that permits the user to review it outside of a social network. Agencies may wish to explore additional methods for providing effective communication such as using text messaging to provide emergency alerts, or using a recorded telephone hotline to provide information to those who have difficulty accessing the Internet. Agencies may also wish to consider building an alternate interface such as in the form of accessible iPhone and Android apps which displays information that is posted to social media channels and allows the user to interact with the agency.

Alternate formats for documents should be provided to users when they are requested. This requires confirmation that a document could be translated into and printed in Braille, enlarged, or provided as an audio recording when needed. Alternate methods for engagement should also be available. For example, if a user is unable to submit a comment, vote in a poll or participate in a real-time chat, the agency should provide a contact that can provide users with alternate methods for submitting information to the agency and participating in agency engagements and activities.

# Using District and Disability Resources

Another effective method to reach people with disabilities is to network with existing disability resources. Often disability –related information is shared by disability-related resources and users with disabilities by E-mail, social media channels, and other channels created specifically for users with disabilities. Examples of disability resources which can be engaged by agencies include Disability.gov and the Job Accommodation Network. Other District government agencies should be engaged and provided with information as well to pass along to their readership. When information is being provided that pertains to people with disabilities, it should be given to District agencies, departments and offices which provide direct services to people with disabilities. These groups can often pass the information onto members of the community or to clients with disabilities. Examples of District offices and departments that should be given disability-related information include the District Department on Disability Rights, Office on Disability Rights, and the Adaptive Services Division of the D.C. Public Library.

# Appendix A – Checklists

Media Type Relevant Section 508 Standards Checklist

Images § 508-1194.21 Software Applications and Operating Systems:

Paragraph (a), (d)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (a), (d), (e), (f)

§ 508-1194.31 Functional performance criteria:

Paragraphs (a), (b), (f) • All images must contain alternative text

• Alternative text for images that add to the meaning of the page must have informative alternative text. Decorative images must contain a null alternative text

• Server side image maps should be avoided in favor of client side image maps

• If the image maps are server side, ensure that there is an accessible equivalent present

Color § 508-1194.21 Software Applications and Operating Systems:

Paragraphs (i), (g), (j)

§ 508-1194.22 Web Sites and Applications:

Paragraph (c)

§ 508-1194.31 Functional performance criteria:

Paragraph (b) • Color cannot be the only means used to identify elements or controls on a page or document. There must be some other method present.

• Color combinations must have a color contrast ratio of 4.5:1 for text less than 18pts or 14pts bolded, or 3:1 for text larger than 18pts or 14pts bolded

Tables § 508-1194.21 Software Applications and Operating Systems:

Paragraph (d)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (d), (g), (h)

§ 508-1194.31 Functional performance criteria:

Paragraph (a) • Data tables must include proper table structure

• Column and row headers must be identified as such

• Column and row headers must indicate a scope of row or col dependent on the related data cell content

• Avoid placing tables within individual cells

• Ensure that related table data is included in the same table

Forms § 508-1194.21 Software Applications and Operating Systems:

Paragraph (d), (l)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (a), (n), (p)

§ 508-1194.31 Functional performance criteria:

Paragraph (a), (b), (c), (d), (f) • Informative text must be placed at the beginning of the form

• Form elements and controls must be laid out in an intuitive order

• Every form element must include a valid label

• When a form is submitted in error, the error message must be placed at the top of the form

• When time limits for form completion are implemented, the user must be given the option of extending their session time

Links § 508-1194.21 Software Applications and Operating Systems:

Paragraph (d)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (a)

§ 508-1194.31 Functional performance criteria:

Paragraph (a), (f), • Link text cannot be generic in nature.

• Links with same text that point to different locations must be avoided

• A skip repetitive navigation area link must be in place and take the user directly to new content

• Links that open various media or document types outside of the current format must include the document or media type in the link text

Frames § 508-1194.21 Software Applications and Operating Systems:

Paragraph (d)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (a), (i)

§ 508-1194.31 Functional performance criteria:

Paragraph (a), (b) • Frames must have a meaningful title

• Ensure that frames do not use absolute sizing

Page Structure and Text Presentation § 508-1194.21 Software Applications and Operating Systems:

Paragraph (c), (d)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (a), (l), (d)

§ 508-1194.31 Functional performance criteria:

Paragraph (a), (b), (f) • Reading order must follow a logical progression

• Proper headings must be used to introduce new blocks of content.

• Headings must be placed in the correct order

• Page titles must be used and be context sensitive

• Avoid using absolute sizing for text

Style Sheets § 508-1194.21 Software Applications and Operating Systems:

Paragraph (b)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (d), (l)

§ 508-1194.31 Functional performance criteria:

Paragraph (b) • Ensure pages are still readable and are not missing vital elements when colors are removed from the browser through the accessibility settings (in IE Tools (ALT+X) --> Internet Options --> Accessibility --> Ignore colors specified on webpages)

• Ensure documents are still readable when style sheets are removed (in IE Page Menu --> Style --> No Styles)

• Avoid absolute units of size and measurement

Embedded Content § 508-1194.21 Software Applications and Operating Systems:

Paragraph (a), (d)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (a), (l), (m)

§ 508-1194.31 Functional performance criteria:

Paragraph (a), (b), (c), (f) • Ensure that when embedded content is included in a page, that there is an accessible link present to download any necessary plugins required to access that content

• Ensure that the embedded content is directly accessible and able to accessed via the keyboard as well as the mouse

• Ensure focus can be placed on embedded media controls, and using MSAA Inspector tools, ensure the proper accessible names for the controls are being rendered

• Provide textual equivalents for live audio or video only content

Animation § 508-1194.21 Software Applications and Operating Systems:

Paragraph (h), (k)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (a), (j), (l)

§ 508-1194.31 Functional performance criteria:

Paragraph (a), (b) • Ensure a method is in place to allow the user to control the animation, in order to review step by step, by including animation controls

• Provide a link or access to non-animated equivalent

JavaScript § 508-1194.21 Software Applications and Operating Systems:

Paragraph (a), (b), (c), (d), (h), (l)

§ 508-1194.22 Web Sites and Applications:

Paragraphs (a), (l), (n)

§ 508-1194.31 Functional performance criteria:

Paragraph (a), (b) • Ensure the same JavaScript functionality is available from the keyboard as well as the mouse

• Avoid forced focus changes that the user is not aware of. The MSAA Inspector tool’s focus tracker can assist in determining if focus is shifting to another location without user control

• Ensure focus moves properly to opened dialog windows

Appendix B – Section 508 Electronic and Information Technology Accessibility Standards

The following Section 508 standards (subparts) taken from the United States Access Board’s website are applicable to Web-based content for DC.gov.

Subpart A – General

§ 1194.1 Purpose

The purpose of this part is to implement section 508 of the Rehabilitation Act of 1973, as amended (29 U.S.C. 794d). Section 508 requires that when Federal agencies develop, procure, maintain, or use electronic and information technology, Federal employees with disabilities have access to and use of information and data that is comparable to the access and use by Federal employees who are not individuals with disabilities, unless an undue burden would be imposed on the agency. Section 508 also requires that individuals with disabilities, who are members of the public seeking information or services from a Federal agency, have access to and use of information and data that is comparable to that provided to the public who are not individuals with disabilities, unless an undue burden would be imposed on the agency.

§ 1194.2 Application

(a) Products covered by this part shall comply with all applicable provisions of this part. When developing, procuring, maintaining, or using electronic and information technology, each agency shall ensure that the products comply with the applicable provisions of this part, unless an undue burden would be imposed on the agency.

(1) When compliance with the provisions of this part imposes an undue burden, agencies shall provide individuals with disabilities with the information and data involved by an alternative means of access that allows the individual to use the information and data.

(2) When procuring a product, if an agency determines that compliance with any provision of this part imposes an undue burden, the documentation by the agency supporting the procurement shall explain why, and to what extent, compliance with each such provision creates an undue burden.

(b) When procuring a product, each agency shall procure products which comply with the provisions in this part when such products are available in the commercial marketplace or when such products are developed in response to a Government solicitation. Agencies cannot claim a product as a whole is not commercially available because no product in the marketplace meets all the standards. If products are commercially available that meet some but not all of the standards, the agency must procure the product that best meets the standards.

(c) Except as provided by §1194.3(b), this part applies to electronic and information technology developed, procured, maintained, or used by agencies directly or used by a contractor under a contract with an agency which requires the use of such product, or requires the use, to a significant extent, of such product in the performance of a service or the furnishing of a product.

§ 1194.3 General Exceptions

(a) This part does not apply to any electronic and information technology operated by agencies, the function, operation, or use of which involves intelligence activities, cryptologic activities related to national security, command and control of military forces, equipment that is an integral part of a weapon or weapons system, or systems which are critical to the direct fulfillment of military or intelligence missions. Systems which are critical to the direct fulfillment of military or intelligence missions do not include a system that is to be used for routine administrative and business applications (including payroll, finance, logistics, and personnel management applications).

(b) This part does not apply to electronic and information technology that is acquired by a contractor incidental to a contract.

(c) Except as required to comply with the provisions in this part, this part does not require the installation of specific accessibility-related software or the attachment of an assistive technology device at a workstation of a Federal employee who is not an individual with a disability.

(d) When agencies provide access to the public to information or data through electronic and information technology, agencies are not required to make products owned by the agency available for access and use by individuals with disabilities at a location other than that where the electronic and information technology is provided to the public, or to purchase products for access and use by individuals with disabilities at a location other than that where the electronic and information technology is provided to the public.

(e) This part shall not be construed to require a fundamental alteration in the nature of a product or its components.

(f) Products located in spaces frequented only by service personnel for maintenance, repair, or occasional monitoring of equipment are not required to comply with this part.

§ 1194.4 Definitions

The following definitions apply to this part:

Agency. Any Federal department or agency, including the United States Postal Service.

Alternate formats. Alternate formats usable by people with disabilities may include, but are not limited to, Braille, ASCII text, large print, recorded audio, and electronic formats that comply with this part.

Alternate methods. Different means of providing information, including product documentation, to people with disabilities. Alternate methods may include, but are not limited to, voice, fax, relay service, TTY, Internet posting, captioning, text-to-speech synthesis, and audio description.

Assistive technology. Any item, piece of equipment, or system, whether acquired commercially, modified, or customized, that is commonly used to increase, maintain, or improve functional capabilities of individuals with disabilities.

Electronic and information technology. Includes information technology and any equipment or interconnected system or subsystem of equipment that is used in the creation, conversion, or duplication of data or information. The term electronic and information technology includes, but is not limited to, telecommunications products (such as telephones), information kiosks and transaction machines, World Wide Web sites, multimedia, and office equipment such as copiers and fax machines. The term does not include any equipment that contains embedded information technology that is used as an integral part of the product, but the principal function of which is not the acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information. For example, HVAC (heating, ventilation, and air conditioning) equipment such as thermostats or temperature control devices, and medical equipment where information technology is integral to its operation, are not information technology.

Information technology. Any equipment or interconnected system or subsystem of equipment, that is used in the automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information. The term information technology includes computers, ancillary equipment, software, firmware and similar procedures, services (including support services), and related resources.

Operable controls. A component of a product that requires physical contact for normal operation. Operable controls include, but are not limited to, mechanically operated controls, input and output trays, card slots, keyboards, or keypads.

Product. Electronic and information technology.

Self-Contained, Closed Products. Products that generally have embedded software and are commonly designed in such a fashion that a user cannot easily attach or install assistive technology. These products include, but are not limited to, information kiosks and information transaction machines, copiers, printers, calculators, fax machines, and other similar types of products.

Telecommunications. The transmission, between or among points specified by the user, of information of the user’s choosing, without change in the form or content of the information as sent and received.

TTY. An abbreviation for teletypewriter. Machinery or equipment that employs interactive text based communications through the transmission of coded signals across the telephone network. TTYs may include, for example, devices known as TDDs (telecommunication display devices or telecommunication devices for deaf persons) or computers with special modems. TTYs are also called text telephones.

Undue burden. Undue burden means significant difficulty or expense. In determining whether an action would result in an undue burden, an agency shall consider all agency resources available to the program or component for which the product is being developed, procured, maintained, or used.

§ 1194.5 Equivalent Facilitation

Nothing in this part is intended to prevent the use of designs or technologies as alternatives to those prescribed in this part provided they result in substantially equivalent or greater access to and use of a product for people with disabilities.

Subpart B — Technical Standards

§ 1194.21 Software Applications and Operating Systems

(a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.

(b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.

(c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.

(d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text.

(e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application’s performance.

(f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.

(g) Applications shall not override user selected contrast and color selections and other individual display attributes.

(h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.

(i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

(j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.

(k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.

(l) When electronic forms are used, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

§ 1194.22 Web-based Intranet and Internet Information and Applications

(a) A text equivalent for every non-text element shall be provided (e.g., via “alt”, “longdesc”, or in element content).

(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

(d) Documents shall be organized so they are readable without requiring an associated style sheet.

(e) Redundant text links shall be provided for each active region of a server-side image map.

(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

(g) Row and column headers shall be identified for data tables.

(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.

(i) Frames shall be titled with text that facilitates frame identification and navigation.

(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.

(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).

(n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

(o) A method shall be provided that permits users to skip repetitive navigation links.

(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

Subpart C – Functional Performance Criteria

§ 1194.31 Functional Performance Criteria

(a) At least one mode of operation and information retrieval that does not require user vision shall be provided, or support for assistive technology used by people who are blind or visually impaired shall be provided.

(b) At least one mode of operation and information retrieval that does not require visual acuity greater than 20/70 shall be provided in audio and enlarged print output working together or independently, or support for assistive technology used by people who are visually impaired shall be provided.

(c) At least one mode of operation and information retrieval that does not require user hearing shall be provided, or support for assistive technology used by people who are deaf or hard of hearing shall be provided.

(d) Where audio information is important for the use of a product, at least one mode of operation and information retrieval shall be provided in an enhanced auditory fashion, or support for assistive hearing devices shall be provided.

(e) At least one mode of operation and information retrieval that does not require user speech shall be provided, or support for assistive technology used by people with disabilities shall be provided.

(f) At least one mode of operation and information retrieval that does not require fine motor control or simultaneous actions and that is operable with limited reach and strength shall be provided.

Subpart D — Information, Documentation, and Support

§ 1194.41 Information, Documentation, and Support

(a) Product support documentation provided to end-users shall be made available in alternate formats upon request, at no additional charge.

(b) End-users shall have access to a description of the accessibility and compatibility features of products in alternate formats or alternate methods upon request, at no additional charge.

(c) Support services for products shall accommodate the communication needs of end-users with disabilities