Section 508 Website Accessibility

for D.C. Government



District of Columbia

Office of Disability Rights

May 2013

Overview

This guide provides basic instruction for making web content that is 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 all users can access them. Also included are recommendations on how to use websites and social media to reach and provide information to citizens with disabilities, as well as a section on Drupal so web content creators can stay informed about various accessibility modules available from third-party “contributed” modules.

The methods and formats for using the web to deliver information are almost limitless. This guide will not present solutions for every way information can be delivered, but it will offer basic concepts and techniques for making information accessible. Web 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. When 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

[Section 508 Website Accessibility 1](#_Toc11068768)

[for D.C. Government 1](#_Toc11068769)

[Overview 2](#_Toc11068770)

[Definition of Terms and Acronyms 5](#_Toc11068771)

[Accessibility Overview 8](#_Toc11068772)

[Disability Types 9](#_Toc11068773)

[Visual 9](#_Toc11068774)

[Mobility 9](#_Toc11068775)

[Speech 9](#_Toc11068776)

[Auditory 9](#_Toc11068777)

[Cognitive 9](#_Toc11068778)

[Age-Related 10](#_Toc11068779)

[Accessibility Standards 10](#_Toc11068780)

[Section 508 10](#_Toc11068781)

[Web Content Accessibility Guidelines 10](#_Toc11068782)

[Assistive Technologies 11](#_Toc11068783)

[Screen Readers 11](#_Toc11068784)

[Screen Magnifiers 11](#_Toc11068785)

[Speech Recognition Software 12](#_Toc11068786)

[Web Accessibility Requirements 13](#_Toc11068787)

[Images 13](#_Toc11068788)

[Color 15](#_Toc11068789)

[Tables 17](#_Toc11068790)

[Forms 18](#_Toc11068791)

[Links 21](#_Toc11068792)

[Frames 23](#_Toc11068793)

[Page Structure 24](#_Toc11068794)

[Lists 26](#_Toc11068795)

[Style Sheets and Styles 27](#_Toc11068796)

[Embedded Content 29](#_Toc11068797)

[Animation 31](#_Toc11068798)

[JavaScript 32](#_Toc11068799)

[WAI-ARIA 33](#_Toc11068800)

[References and Tools 35](#_Toc11068801)

[Automated Testing Tools and Plug-ins 36](#_Toc11068802)

[Color Testing 36](#_Toc11068803)

[MSAA Testing Tools 36](#_Toc11068804)

[PDF Accessibility Tools 36](#_Toc11068805)

[Section 508 Guide and Best Practices 36](#_Toc11068806)

[Screen Readers 36](#_Toc11068807)

[Data Tables 36](#_Toc11068808)

[Drupal 37](#_Toc11068809)

[Overview 37](#_Toc11068810)

[About Modules 37](#_Toc11068811)

[Core Modules 37](#_Toc11068812)

[Requirements 37](#_Toc11068813)

[Contributed Modules 38](#_Toc11068814)

[Page Style 39](#_Toc11068815)

[Text Size 39](#_Toc11068816)

[Accessibility Helper 39](#_Toc11068817)

[HTML Purifier 39](#_Toc11068818)

[Readability Analyzer 39](#_Toc11068819)

[Choosing Contributed Modules 39](#_Toc11068820)

[Social Media Accessibility 41](#_Toc11068821)

[Overview 41](#_Toc11068822)

[Social Media Networks 41](#_Toc11068823)

[Facebook 41](#_Toc11068824)

[Twitter 42](#_Toc11068825)

[YouTube 43](#_Toc11068826)

[User Impact 44](#_Toc11068827)

[Requirements 45](#_Toc11068828)

[Images 45](#_Toc11068829)

[Video 46](#_Toc11068830)

[Widgets and Third-Party Sharing 47](#_Toc11068831)

[Reaching Constituents with Disabilities 48](#_Toc11068832)

[Using Websites for Outreach 48](#_Toc11068833)

[Using Social Media for Outreach 49](#_Toc11068834)

[Using Blogs and Newsletters 49](#_Toc11068835)

[Providing Alternate Formats and Methods of Presentation 50](#_Toc11068836)

[Using District and Disability Resources 50](#_Toc11068837)

[Appendix A – Checklists 51](#_Toc11068838)

[Subpart A – General 56](#_Toc11068839)

[Subpart B — Technical Standards 59](#_Toc11068840)

[Subpart C – Functional Performance Criteria 61](#_Toc11068841)

[Subpart D — Information, Documentation, and Support 62](#_Toc11068842)

# Definition of Terms and Acronyms

**A**

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

AT**:** Assistive Technology - A term that includes assistive, adaptive, and rehabilitative devices for people with disabilities that provide enhancements to or changed methods of interacting with technology.

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**:** An attribute of an image that uses text to describe an 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 are multiple, independently controllable sections on a Web presentation. Building each section as a separate HTML file and having one “master” HTML file identifying all of the sections achieves this effect.

**G**

Graceful Degradation**:** The property that enables a system to continue operating in the event of a failure of some of its components (e.g. ARIA is not supported with particular browser and assistive technology combinations).

**H**

HTML**:** Hypertext Markup Language-the main [markup language](http://en.wikipedia.org/wiki/Markup_language) for creating web pages and other information that is displayed in a [web browser](http://en.wikipedia.org/wiki/Web_browser).

**I**

IE**:** Internet Explorer (browser)- a series of graphical web browsers developed by Microsoft and included as part of the Microsoft Windows line of operating systems.

IFrame**:** Inline Frame - An IFrame 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 are interpreted by the application.

**L**

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

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

List Item**:** Blocks of text or content that are grouped with other related content and 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 convey content to non-sighted users based on a webpage’s source code, PDF tag trees, and document content.

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

**W**

WAI**:** Web Accessibility Initiative – A group of W3C members that develop strategies, guidelines, and resources to improve Web accessibility for 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

D.C. government departments are continuing the transition from delivering programs and services through traditional paper-based formats to providing services electronically. It therefore 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. The standard definition of accessibility in this context concerns the degree to which information, services, and the physical environment are 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. Some issues for users that need to be considered include:

* Users should always have an adjustable scale option on each web page, whether they are accessing information through a computer, tablet, mobile phone, or other device
* Users should be able to control web interfaces, whether they are using pointing devices such as mice, keyboards and keyboard emulators, touch screens, or speech recognition software
* Users should be able to change text color or text size on a webpage
* Users should be able to read the web page using speech synthesis or refreshable Braille
* Users may need captions for audio content, or if they require language that is easier to understand.

Universal design means making technology available to all users in as many contexts as possible. Site developers and designers should strive to use universal design principles as much as possible. The web site www.usability.gov provides additional information on the topics of usability and universal design.

The DC.gov Accessibility Policy implements the accessibility standards federal government agencies are required to meet to make their technology accessible to people with disabilities. The Section 508 Electronic and Information Technology (EIT) standards, which are part of the 1998 Rehabilitation Act, include requirements that all webpages on the DC.gov portal be universally accessible. The District of Columbia is not a federal government establishment, so Section 508 does not apply directly to the D.C. government. However, Title II of the 1990 Americans with Disabilities Act (ADA) 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. The 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 for people with disabilities as they are for others, the D.C. government has adopted Section 508 requirements as a minimum standard that must be met for all web communications.

## Disability Types

### Visual

Individuals with visual disabilities lack the ability to see or have difficulty with sight. Visual disabilities include blindness, reduction of visual acuity (visual range), color blindness, and tunnel vision. The adaptive technologies used on the web by people facing visual challenges vary 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 that converts information into speech synthesis or refreshable Braille.

### Mobility

Individuals with mobility disabilities are those that have 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 that emulates a keyboard or pointing device such as a head pointer. Others may use software solution such as an on-screen keyboard or speech recognition software.

### Speech

Individuals with speech disabilities lack the ability to speak or have difficulty producing speech. They often use augmentative communication devices that 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 to hear 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 cognitive processes. Cognitive disabilities cover 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 that 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 reported that 37 percent of people 65 and older have at least one disability. Web use and computer penetration among the population aged 65 and older is growing. Today the “baby boomer” generation makes heavy use of the web.

## Accessibility Standards

### Section 508

Section 508, as stated above, is a U.S. federal law requiring electronic and information technology that is developed, purchased, used, or maintained by the federal government to be accessible to people with disabilities. It also 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: 1) general requirements, 2) technical standards, 3) functional performance criteria, and 4) information, documentation, and support requirements.

The “technical standards” category is further divided: 1) software applications and operating systems, 2) web-based information and applications, 3) communications products such as telephone and voice mail systems, multimedia and video products, 4) self-contained systems such as kiosks and office equipment, and 5) 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 used as the basis of domestic web accessibility regulations and standards in many countries. The current Section 508 standards are based on version 1.0 of the 1999 Web Content Accessibility Guidelines. 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 strive to meet the requirements of WCAG 2.0 level AA conformance, which goes beyond the Section 508 requirements. The WCAG 2.0 criteria are more detailed than the Section 508 requirements, and are therefore likely to produce a more accessible experience for users with disabilities. 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. 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, even the W3C 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 software, devices, or techniques used to aid individuals with disabilities in the performance of tasks that 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 recite the contents of the computer screen aloud using speech synthesis or optionally through 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 by 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 on the screen, allowing users with low vision to read information and see images more clearly. Magnification levels from 1.2 to 16 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 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 and enter text into an application as if it had been typed using a 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. 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.

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).

# 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. 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, as well as 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 that are embedded in a document. For most web pages, the image content is contained in an external file and then is embedded into a page by referencing the file location in the HTML for the webpage. The file only contains information required to form the shapes and colors that make up the image, and does not contain any text that could be interpreted by a computer, browser, or assistive technology. If images are used to provide information that contributes to the understanding of the page, users who cannot see the image will lose the meaning that the image represents. 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 that reverses the colors used on the screen. If an image is part of a link, voice recognition software will speak the alternative text provided for an image to gain access to the link. For the reasons listed above, it is important that alternatives such as text 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 that 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 information will not be provided about it.

Concise and informative alternative text should be provided for meaningful images. 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.” 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 users who have difficulty understanding or seeing the images can benefit from the descriptions.

Decorative images are those that are used for visual effect and do not assist with 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. 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. 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.

An image map is an image containing multiple active “regions” within the image that 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 takes 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).

**Resources**

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. This 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 on how 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 U.S. 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 U.S. 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 is to read the page with a screen reader, which will ensure that alternative text is provided for images and that image maps can be accessed using the keyboard. 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 that are missing meaningful text alternatives. 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.

## Color

Color will often be used to indicate the state of items on the page, such as identifying items that have been selected, or identifying form fields that need to be corrected due to errors. Color can also be used to tell items apart, like in 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, and users with colorblindness may be unable to tell colors apart. Users with low vision may be unable to tell the difference between the foreground and background, making content difficult to see. For these reasons, color and visual formatting should be used in ways that do not prevent users from obtaining the information on a webpage.

**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 requirement does not prevent the use of color to enhance the identification of important features. However, it does require that some other method of identification be combined with the use of color. This requirement does not apply to color or other visual formatting used as decoration or for visual appeal.

Color contrast refers to the degree of difference between the foreground (used to display text, images, controls, and other page content) and background colors (the color against which items on the page are presented) used on a page. When foreground and background colors on a page are too much alike, users with low vision or color blindness 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 that is included as part of an image. Color pairings for text and for images that contain text must have enough contrast 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. According to the WCAG 2.0 requirement, text that uses less than an 18-point font or uses less than a 14-point bolded font must use a color contrast ratio of 4.5 to 1 or more. Text that uses an 18-point or larger font, or text that uses 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.

**Resources**

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 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 contrast can be tested using the eyedropper tool to obtain the numeric values representing the foreground and background colors on a web page. One such tool is the Contrast Analyzer for Windows and Mac from the Web Accessibility Tools Consortium and the Paciello Group. 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 display relationships between pieces of information using formatted rows and columns. Tables that contain row or column headers (or both) that identify how all the data in a particular row or column is related is called a “data table.” Tables that use rows and columns in order to present a particular visual appearance, but are not used to organize related data, are called “layout tables.” According to OCTO requirements, layout tables 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 have difficulty moving around the table and understanding relationships between data within the table. They are unable to link the content of individual cells with the column and row headers, making the information effectively useless.

**Requirements**

Tables that relate pieces of information and are laid out as a grid containing rows and columns must be coded as a data table so an adaptive technology can correctly identify individual tables, rows, columns, cells, column headers, and row headers. It also means that barriers must not exist that would prevent review, movement or understanding within a table. For example, placing tables inside each other (nesting tables) or merging cells together can create difficulty moving between cells or maintaining the relationship between the cells and the row/column headers, thus creating a barrier. 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.

Row and column headers must be labeled as such using coding techniques. Changing the visual appearance using bolding, different font size or other changes in visual appearance, are not identified differently by assistive technology. This increases the difficulty in understanding the purpose of each table cell. This also means that coding techniques used to identify row and column headers must not be used on table cells that are not actually headers. Table headers and cells must be located in the same table and not separated by any barriers in order for the relationships to be understood by adaptive technologies.

Additional information about a table should be provided if doing so would help to better convey the information in 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 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.

**Resources**

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 must be provided to 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. One such tool is the HTML Table Validator, found on the Web Experience Toolkit site. Another useful way to test for data table accessibility is to review the table using a screen reader. 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 that makes up the table.

## Forms

The ability for all users to successfully fill out online forms is critical to ensuring access to the usability and functionality of many websites and web applications. Users with disabilities can face major challenges in filling out and submitting online forms that are not properly constructed. Users can struggle with locating form instructions; determining what should be entered into each form field and how that information should be formatted; determining what fields are required; recovering from field input errors; and time limits on form submission. It is important not to create barriers that prevent users from completing and submitting online forms.

**Requirements**

Form instructions must be placed at the beginning of a form. Form instructions should be clear and describe any important information needed to complete the form, such as how “required fields” will be identified on the form.

Form controls must 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. An example of out-of-sequence form control is if a form requests a person’s middle name before requesting the first name, or inserts 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 must be associated with a label. When form controls are tied to their respective labels, screen readers will announce the label text when the user reaches the related control. Additionally, 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 all information needed to complete the form field, such as informing the user if the field is required, the format in which entry should be completed, or password length and character requirements. Furthermore, label text for a form control must 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.

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 must be placed at the beginning of the form. The message should allow the user to easily locate the field containing the error and it must clearly describe the changes that need to be made. Methods that can aid the user in locating and correcting errors 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 that require the user to fill in a form 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. Roughly one minute before the user is about to be logged out of their session, provide a warning that time is about to run out and give the option for the user to continue their session with a simple 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.

**Resources**

Resources 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 in the Forms section of the SSA’s 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.

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 Martin Luther King Jr. D.C. Public Library.

**How to Test**

Forms can be tested using the keyboard to verify that they are keyboard-accessible. Testing a form’s 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 should cause the form control(s) to become active.

Many automated tools can scan the source code for missing labels and other indicators that impact accessibility. Another good way to verify that forms are accessible is to test them using a screen reader. A screen reader will include commands that will bring up all form controls into a list. This not only allows one to easily locate and move between form controls, but it also allows the tester to visually confirm what the screen reader will announce as the label for each control.

The only way to be certain that forms have been properly coded is to manually review the code.

## Links

The displayed text of a link helps users of assistive technologies determine what will occur if the link is followed. A user may do this when reviewing the link in the context of surrounding text or when moving between links and reviewing the link text out of context from surrounding text. If link text is not provided, is vague, or does not adequately 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 the link is useful. Similarly, if the same link text is used for multiple links on a page that go to different locations or perform different actions, the user may be unsure which link to follow. Using the text “click here” or “read more” for many links on a page with different locations is currently prohibited by OCTO standards.

Incorporating 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 unless a method is provided to skip them. Overuse of links can make browsing slow for a user who moves to links using the tab key, or for a screen reader user who hears the page read in order from beginning to end.

**Requirements**

All links on the page must provide clear text that describes the purpose of the link. For links that take the user to a different location, the link should identify the title of the location. 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. The user may review a page by tabbing through all links on the page, and 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 that 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 that 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.

Links should be provided to skip over items that 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 announced every time a new page loads. A method should be provided to allow users to skip to new content.

If the link leads to a file other than a web page, the link should state the file type and relative size so 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 users 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.

Section 508 standards require that for non-web page files, only accessible files and applications may be posted to websites. Additionally, 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 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, such as a text description of an image.

Links should respond to keyboard actions. Users who are blind or who have limited mobility may use the keyboard to browse web pages. Providing links that only respond to the mouse may cause challenges for such users. 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.

**Resources**

While Section 508 does not have any direct requirements regarding 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 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 SSA 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 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.

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 using the keyboard to confirm that they can be reached using the tab key and that they can be accessed using the keyboard. Links 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 that is intentionally not shown on the screen is provided for screen readers to aid with understanding the purpose of the link.

If a link or other method is provided to skip past information that is repeated on every web page, this method should be tested using the keyboard. That link should be the first link on the page that the user can reach with the tab key after 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 that 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.

## 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 a page must have a descriptive frame title and name. A descriptive title gives the user information that will assist them in determining the frame’s content. Users are able to navigate to specific frames on a page, and providing appropriate titles allows users to make accurate choices. Also, many screen readers have the capability to navigate among the frames using the CTRL+TAB and CTRL+SHFT+TAB keystroke combinations, so proper frame structure and descriptive titles will allow visually impaired users to navigate effectively.

When frames or iframes are used on a 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 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 need to be resized.

**Resources**

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**

A tester should test frames, their appearance, and their structure using screen reading software while tabbing through the frames. When a frame is in focus, the screen reader will be able to read the name and the title on the frame.

Frames can also be tested by manually inspecting the source code of the page or document to identify the frames used and associated information.

## 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. Visually impaired users who use screen readers rely on appropriate page structure to render the proper reading order of information and relay informative elements such as the page title, headings and list items.

**Requirements**

Because visually impaired users may be unable to benefit from 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’s or document’s contents. Normally, assistive technology reads page content in the top-to-bottom and left-to-right reading order. Deviations from that order must be properly coded in order to be recognized by the assistive technology. For example, a visual heading on a page cannot be utilized as a heading until the heading is properly coded. This includes, but is not limited to, ensuring the appropriate tags are used, placing tags in the correct order, and eliminating unnecessary tags.

Proper page structure and heading structure on a page or document allows users of assistive technologies to access the information easily and assists users of screen readers with page navigation. The developer should also use meaningful titles for pages within a website or document because the page title informs users about the nature of the content that the page delivers. Without proper page titles, users who visit a specific page in a website or document cannot be sure they have arrived at the correct location without having to investigate the page’s content.

Also, the text on the page should be resizable. That is, a user of the page should be able to increase or decrease the size of the text on the page using the browser’s text re-sizing tools. Developers should use relative sizing and avoid the use of absolute sizing to allow for this adjustment.

**Resources**

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

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

**How to Test**

A tester can verify that headings are accessible using a screen reader. When the tester navigates to a heading on a page or in a document, the screen reader should identify the heading as such, and should read the appropriate information to the tester. Screen readers with more advanced options allow for the user to pull up a heading list, and to navigate from one to another.

A tester can also verify that headings are present by manually inspecting the source code of the page. A tester should be looking for proper <h> tags in HTML source code. When working with documents, such as PDFs, a 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. When working with Word documents, a tester can go to View and check the Document Map checkbox to view the heading structure of the document and verify that headings are being used properly.

A tester can verify that page titles are being used correctly by viewing the top of the browser bar. Viewing the information displayed there will tell the tester that a page title exists and what that title is.

## 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. 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 that they are related items. For example, when developers use graphic elements such as images of bullets next to related text, the items will be grouped together visually, but screen readers will not transmit that relationship to users. Also, using this method will not notify users of the beginning and end of the list, nor the number of related items in the list. Using list item structure enclosed in proper ordered or unordered list HTML containers will ensure that the appropriate relationship is conveyed to users.

**Requirements**

All related content on a page presented visually as a list must be structured as a list, in the page’s source code or in the 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 hierarchical 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 all nested lists fall within the same list container.

**Resources**

Using appropriate list item elements and placing them within list containers in the proper order is vital to ensuring that hierarchical relationships of content are exposed to the user in the correct manner. Informative and technical information regarding list structure can be obtained 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), the list structure conveys. More information on creating lists in Microsoft Word, can be obtained at 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. More information on creating ordered and unordered lists in HTML can be found on the W3 Schools website.

**How to Test**

A tester can verify that lists are properly structured and accessible using a screen reader. When testing with a screen reader, a tester 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. When the tester reaches the end of the list, the screen reader will announce the end. These announcements will validate that the list is, at a minimum, enclosed within appropriate list containers. To ensure that all list items have been properly structured, a tester can verify that the number of list items announced by the screen reader corresponds to the number of visual list items present on the page. A tester can 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

Style sheets help developers determine or change the way information is displayed on web pages. Style sheets can be used to present text in different colors and fonts, to place content such as blocks of text in specific locations on a page, and to display elements such as background images. Styles are used when working with documents (e.g., Microsoft Office documents). Content editors can utilize built-in styles to control how the end user views the documents.

Several accessibility issues can arise when using style sheets or styles. The first problem arises when pages are dependent on style sheets to expose certain content. For example, 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 corresponding with the background images), low vision users who disable colors on a page will not be able to see those elements. A second problem is tied to the way that text is sized. When relative units such as percentages are used to size the text, users with disabilities such as low vision and color-blindness are able to increase (or decrease) the text size through the browser or the document-rendering software. However, when pixels and points are used to size the text, users may be unable to adjust the size of the content to meet their own needs or preferences. Third, care must be taken to ensure that documents retain appropriate styles when exported. If style settings are not created at the source document level, they will not transfer over to exported files such as PDFs.

**Requirements**

Users must be able read web pages correctly even when style sheets (or, at a more basic level, color) are removed. If style sheets are removed, users must still be able to determine the correct reading order and all active elements on a page must be visible with at least a textual alternative. Users must also be able to utilize their own external style sheets from within the browser’s built-in accessibility menu in order to override the default style sheets. Pages must allow users to choose their own style settings so they can tailor page content to their particular needs.

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

**Resources**

Developers must follow certain rules to ensure the accessibility of web pages that utilize style sheets or documents that use styles. When creating web pages, developers must ensure that pages can be viewed correctly when style sheets are removed and must avoid using absolute sizing as this will prevent users from adjusting the content on a page through magnification. When creating documents, content editors 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. Using the styles menu, which can be found under the Home tab of Microsoft Office, will ensure that the styles remain when the document is converted to secondary software such as Acrobat PDF.

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

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

Techniques and information about hidden content on web pages 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 tester can test that style sheets are being used in an accessible manner by disabling and re-enabling the attached style sheets on the browser’s View Menu [in Internet Explorer go to View – Style – No Style (or Default Style)]. By disabling the style sheets, the tester can confirm that logical reading order is correctly in place because 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 still displayed on the page.

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

The tester can also ignore colors through the browser’s Accessibility menu (in Internet Explorer go to 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 through the browser’s View menu (in Internet Explorer go to View – Text size - larger).

## Embedded Content

Media content embedded in a document or on a web page raises several accessibility issues. First, developers and content editors must ensure that users can directly download programs or plug-ins necessary to access embedded content. In addition, embedded content must be directly accessible. For example, setting the Flash content’s wmode to window will allow assistive technologies to access the content and its controls. Also, textual equivalents are important for live audio or video-only embedded content. This can take 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 embedded 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 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 is present on the page, a direct link to download the required program or plug-in also must be present on the page. Such a link helps users with disabilities to download the plug-in immediately without leaving the page or location containing the embedded content.

Users with disabilities also must 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 of and, when possible, a textual alternative to the embedded content should be present on the webpage or the document. The embedded content must be accessible before it is embedded on the webpage or other media. For example, Flash video can be made accessible in the Flash authoring environment before the video is embedded on the webpage. If embedded content 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 users with disabilities are able to read the presentations. For video-only presentations, the captioning of the video must be available in an accessible manner so 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.

**Resources**

Information regarding 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 multimedia presentations and text alternatives to such 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 accessible Adobe Acrobat PDF documents 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 a website can be found in the Creating Accessible Flash Course on the U.S. Department of Veterans Affairs Website.

**How to Test**

A tester can test that a direct link is available to download the plug-in or program necessary to access embedded content by visually scanning the page or document in question and ensuring that the required link is present. 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.

A tester can verify that Flash content (i.e., a Flash Video) is accessible by accessing it using a keyboard and a screen reader. When the tester navigates with the Tab key, the Flash content should gain a keyboard focus. When the tester navigates to the Flash contents with the Tab key while using a screen reader, the tester should be able to hear the text equivalent of the Flash contents.

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

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 helps track 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.

## Animation

Developers should keep several accessibility issues in mind when content incorporates animation and animated effects. Developers must consider 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 or be unable to extract information without repeated viewings. 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 link that takes the user to a site like Time.gov to allow the user to check the time in an accessible format.

**Requirements**

Developers must provide meaningful 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 stories.

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

**Resources**

For more information about accessible animation controls, visit the U.S. Department of Veterans Affairs “Creating Accessible Flash Course.”

**How to Test**

Testers may verify that when animation appears on a page or in a document, controls are included to 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 are present.

Visual inspection of the document in question can reveal if a link has been provided that allows the user to access an accessible equivalent to the animation. Ensure this link can be activated with the keyboard and that its location 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 (such as form field data), to insert content on the page dynamically without reloading the page, and to trigger events in response to a user’s actions, such as mouse clicks and keyboard presses. A JavaScript event is an action on the web page that occurs after a user has done something on the page to cause the JavaScript to start.

One powerful feature of JavaScript is the ability to bring focus to a location on the page. For example, JavaScript may create a blinking cursor on a form field or a dotted rectangle around a link or button to highlight an area of the web page that needs input or attention from a user.

**Requirements**

The most important requirement for making a page with JavaScript accessible is to ensure that a user can use the page with the keyboard. Often, mouse movement or actions trigger JavaScript events, with no consideration for keyboard alternatives.

Avoid focus changes that a user may not 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 at 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.

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 also occur when the keyboard is used.

A button must be 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.

**Resources**

Additional information on JavaScript accessibility can be found in the Creating Accessible JavaScript article on the Web Accessibility in Mind (WebAIM) website.

**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 user presses the spacebar or Enter key on a button or link.

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 be able to open a combo box with Alt+Down Arrow without a page refresh or change of focus. A button must be present next to a combo box that can be accessed 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. It works specifically with screen reading devices for users who 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. ARIA can be used to identify the “role” of “progress bar.” Next, providing the object a label such as “loading” can identify the “name.” Setting the appropriate ARIA values, often in a percentage such as 80%, can provide the current value.

As the Internet is becoming richer and more interactive, many developers create custom form fields like checkboxes to make forms more visually appealing. The challenge is that some browser and assistive technology combinations do not support ARIA. The “state” of a checkbox, for example, can be “checked” or “unchecked” at any time. If a custom checkbox is created, the state cannot be provided automatically to ARIA . 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 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, checkboxes, 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.

JavaScript is an integral part of ensuring that the names, states, values, 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.

**Resources**

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 behavior 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

# Drupal

## Overview

Drupal is a content management system (CMS) that gives great flexibility for website design and content creation. It is the content management system used by all D.C. government websites. Drupal is becoming one of the most popular ways to build and add content to a website. It is estimated that about one quarter of .gov websites use Drupal to build and update websites, including WhiteHouse.gov and Commerce.gov. These websites must also comply with Section 508 requirements, and Drupal has many options for accessibility.

Content management systems are used because they allow non-tech savvy people to easily add content, such as blog posts or images, to websites through simple interfaces. Other popular content management systems include WordPress and Joomla!

### About Modules

In Drupal, content is updated and added through modules. Modules can help users post blogs and articles, add images to galleries, set up polls for online voting, create discussion forums, and much more. Modules are set up by site developers to work with site themes and designs. For example, developers may set up an area of a Drupal-based website for images, but content authors may be the ones adding images to the gallery.

Drupal comes with a set of core modules, but the Drupal community is very active in creating third-party “contributed” modules to extend customization of sites. These third-party modules can easily be downloaded and installed for free, and can greatly help increase the accessibility of a website for people with disabilities.

## Core Modules

Drupal comes with many core modules that assist users in creating common website content. As stated in Drupal’s “Accessibility Statement,” available in the “About” section of the Drupal website, core modules in Drupal 7.0 were created to conform to WCAG 2.0. However, users must be aware that accessibility can only be achieved when the proper fields are included during content addition and creation. Fields that are vital to accessibility, such as alternate text for images, are not necessarily required to add content to Drupal.

As new versions of Drupal are regularly released, it is important to test website accessibility after modules are updated or new modules are installed. This is the best way to ensure accessibility.

### Requirements

Images posted to Drupal sites must include alternate text, which can be read by screen readers. Text should be as descriptive as possible and detail what is in the image for users who cannot view it. When adding images through Drupal, a field for adding alternative text comes up after the image is uploaded. Content creators must be certain to check for this field and add alternative text once the upload is complete and before saving content to a site.

Content creators can control the color of links, text and backgrounds through the Color module. For users with low vision or color blindness, contrasting colors are vital to conveying text information. For best accessibility, content creators should ensure that text has a high color contrast to help ensure visibility. Websites with good color contrast are also readable in grayscale.

Contact forms can be created through core Drupal modules, and these need proper form labeling to ensure they can be read and understood by users of assistive technologies like screen readers. Labels on forms must be clear and state what information needs to be filled in. Additionally, forms must move logically through subjects and topics. For example, a form should ask for a first name before a middle name, and name fields should not appear in the middle of an address field.

Core Drupal modules allow content creators to add pages to website menus as a way to better organize information. For these menus to be accessible, they must include a description that details what information can be found in the menu. Drupal provides a “Description” field for this purpose. When a description is included, a screen reader can read it. Additionally, the description shows up as text when a user hovers the mouse pointer over the menu

## Contributed Modules

Major strengths of Drupal include the endless customization options and a strong, supportive user community. These aspects have resulted in thousands of contributed modules, which are developed and maintained by parties outside of Drupal but are shared freely online for Drupal users. Drupal.org and DrupalModules.com are both websites that can be used for searching and reviewing modules. When encountering accessibility errors and barriers on a Drupal website, modules can be a quick and easy way to fix problems.

It is important to read through reviews of Drupal modules before installing and make sure modules will work with current the Drupal version. Additionally, modules should be tested for accessibility, and caution should be taken with modules that are not expressly developed to assist in accessibility.

New modules are constantly being created, and it is impossible to highlight all useful accessible modules. However, there are many popular accessibility modules that have proven over time to be highly effective and useful for websites that conform to Section 508 and WCAG 2.0 standards.

### Page Style

Page Style is a module that changes the appearance of a website by overriding CSS style sets. Page Style adds new page appearance options, giving users a choice to switch the page to black text-on-white background or a white text-on-black background view. These options are helpful to users with certain kinds of vision disability. The module inserts a menu for changing page appearance onto its home website and does not use JavaScript.

### Text Size

The Text Size module provides an option for users to increase or decrease text size to better see a page. In addition to changing text size, the module also automatically changes the size of objects on the page, such as menu boxes. This maintains the readability of the text and the overall appearance and styling of a page. Text Size adds a menu onto a page with its resizing options and does not require JavaScript to run.

### Accessibility Helper

Accessibility Helper is a collection of modules that work together to provide options to enhance the accessibility of core Drupal modules and provide a toolset to fix modules with known accessibility errors. This module is aimed primarily at Drupal developers who are responsible for overseeing maintenance of Drupal modules.

### HTML Purifier

A popular module for cleaning up malicious HTML code, HTML Purifier also helps content contributors make sure their content complies with WCAG 2.0 standards. HTML code that conforms to these standards is more likely to be understood correctly by screen readers and other HTML-dependent assistive technologies.

### Readability Analyzer

The Readability Analyzer module runs through text content on a Drupal website and gives a grade level reading equivalency. Text that is a high grade level is harder for users with cognitive disabilities, as well as English language learners, to understand. Once content is analyzed, text can be edited for clarity.

## Choosing Contributed Modules

There are several websites to help Drupal find and choose useful Drupal modules. The Drupal community is extremely active in creating, rating and maintaining modules. The Community section of Drupal.org, the official Drupal website, has links to user groups, forums, events, and blog posts for Drupal users to find support and assistance in choosing modules.

Drupalmodules.com is a repository of Drupal modules with ratings, reviews and overviews of contributed modules. These ratings, reviews, and overviews enable users to choose modules with some confidence about their usability and accessibility.

The best modules are up-to-date, actively maintained and functional in the newest version of Drupal. Look for modules that have been updated recently. This indicates active support and maintenance of the module. Both Drupal.org and Drupalmodules.com include release dates for modules. Additionally, they rank module releases by version capability. Release notes on modules will detail what versions the module runs best on. Modules that are only in development phase, and not officially released yet, are not ready to run on active websites.

Even up-to-date and actively maintained modules may still have bugs and errors in need of patching. Drupal users can view upcoming patches on Drupal.org by navigating to a module page and choosing the “View pending patches” section link under the site’s “Development” section. Here, users can see known issues and track what patches are in development or need development. Modules with active patching processes are more likely to be well-maintained and supported in future versions of Drupal. Pay special attention to patch summaries, and look for issues that would conflict with Section 508 requirements or create barriers to accessibility. It is important to remember that even modules needing patches may still be useful and usable. Bugs that do not have an impact on accessibility or usability may be minor enough to warrant module installation.

# 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 a majority of respondents use Facebook and Twitter, and a significant number use YouTube and Flickr. The audit also found a majority of government agencies use social media for promotion and constituent services, and a large percentage use social media for advocacy. Social networking has proven to be an effective avenue for engaging and reaching the public, including in the District.

D.C. government agencies also need to ensure that they are also providing effective communication to people with disabilities through social media. Many social media platforms present accessibility challenges that prevent users with disabilities from fully using and participating in social networks.

With a better understanding of how people with disabilities may use DC.gov websites and social media channels to interact with District agencies, websites and social media can be made accessible and communication can be effectively provided to people who have disabilities.

When faced with inaccessible social networks, some users instead connect to networks through alternate interfaces. These alternates include third-party client interfaces, smartphone apps, or versions of the website that are intended for mobile devices. When posting information to social networks, be aware that users may not be connecting through the standard social network interface, and provide information in ways that can be understood regardless of the device or interface being used.

## Social Media Networks

Facebook and Twitter are the most popular social networks, and most commonly used by organizations to connect with the public. Many organizations will also use platforms such as YouTube to host videos, which can in turn be shared on social networks like Facebook and Twitter. As use of social networks continues to grow, new networks will likely become popular. It is impossible to predict the accessibility issues future social networks may have, so only today’s most popular networks are examined here.

### Facebook

Facebook is a free, interactive social networking platform that allows users to connect with 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 college students with each other, it is important to note that Facebook usage is increasing among the 55-64 age group, even approaching the same level of usage among 13- to 17-year-olds. Recognizing Facebook’s greater penetration within the “baby boomer” generation, organizations have begun using it as a platform for outreach and engagement with customers and clients.

#### Facebook Accessibility

Many people with disabilities use Facebook to access information. The primary Facebook website contains accessibility problems that may change as parts of the interface undergo redesigns. Users of assistive technologies may not have access to certain features on the site, or they may have 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. The user may have to be using updated, recent versions 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 version of the Facebook site intended for mobile devices, m.facebook.com. The mobile interface does not contain as many features as the main site. Some users of screen readers prefer it 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 through the Facebook-developed app or third-party apps that can access Facebook. Reviews of the compatibility of social networking apps with the Voiceover screen reader on 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 called “tweets.” A tweet is a post containing no more than 140 characters of text that is displayed on the profile page of the author and delivered to the author’s subscribers, known as “followers.” Twitter is generally used to provide updates in real time, share information and resources, engage in online discussions, and promote events and activities.

Twitter also provides a search feature that allows users 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 a topic that is prefaced by the “#” symbol. For example, many tweets on the topic of accessibility contain the hashtag “#a11y” to represent accessibility, shortened as “a11y” (there being 11 letters between the “a” and the “y” in the word accessibility). Hashtags do not need to be shortened in this way, but it is not uncommon to save space due to Twitter’s 140-character limit.

Other conventions adopted by users of Twitter include forwarding a tweet on to 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.

#### Twitter Accessibility

The text-based nature of Twitter makes it one of the most accessible forms of social networking, but many accessibility issues exist with the Twitter website’s interface. These include a confusing layout structure, lack of keyboard access, and inability to change text size. To work around these issues, many assistive technology users will navigate Twitter through alternate interfaces or third-party clients.

Some users with disabilities access Twitter through its simplified website developed for mobile devices, m.twitter.com. Others use Twitter apps specially designed for their smartphones. In addition, several non-visual Twitter clients have been created for Windows users of screen readers. Examples of these include The Qube, Klango Twitter, and Twitmonger.

One web application that was developed to provide alternate access to Twitter through an interface that meets WCAG 2.0 requirements is Easy Chirp. Easy Chirp was designed to provide keyboard accessibility, support for screen readers, and a more intuitive layout to the web interface.

### 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 Google brought to YouTube were video captioning and auto-transcription. Vital to accessibility, these features allow video creators to add closed captions to their videos using a combination of experimental speech recognition, automatic timing, and subtitling technologies.

#### YouTube Accessibility

The addition of captioning features is one example of how accessibility of YouTube has improved since Google purchased it. 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 a lack of compatibility with screen readers on platforms such as Mac OS.

Alternate interfaces to the standard player created through JavaScript or HTML5 can be more accessible. These alternatives 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, such as www.facebook.com, www.twitter.com, and www.youtube.com. A review of standard social media interfaces for the most popular platforms as of 2011, can be found through 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 that may be easily solved by a human, but stump a computer. Usually this challenge involves recognizing letters, numbers, or words embedded as part of an image and typing the displayed text into an input field. Because the text is part of an image, screen readers are unable to read the text displayed, while 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 is 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 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 feature that does not allow keyboard access is tagging people in a Facebook photo or post. Screen reader users would find it difficult to tag people because Facebook does not provide keyboard access for 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 videos uploaded to the platform. This leaves the author searching for alternative solutions. Even when the platform does provide alternatives, the alternative must still convey the same level of understanding to a user as the regular content. For example, the YouTube auto-transcription service can make mistakes recognizing speech in a video that must be corrected before the transcript or caption file is published.

Social media accessibility must also be considered outside of social media websites. Widgets and similar components used on websites to display content from or upload content to social media services can also present accessibility issues. These widgets are 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. While social media is a great tool for reaching and engaging the public, there will be those who find it difficult or 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

Images that help users better understand 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

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 that 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.

#### Twitter

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 makes it impossible to provide a text description for the image. Instead, post a link that takes the user to a web page, allowing the user to access both the image and the text description of the image.

### Video

Posts that contain only audio or only video must include 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 user listening to or viewing the file would have. Other than YouTube, most social networks do not support the uploading of transcript files along with media. In such 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.

#### Captioning

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. 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.

#### Audio Description

Audio description, also known as video description or descriptive video, is a verbal description of the key visual elements of a program that is inserted during natural pauses in the program’s dialogue. Key visual elements are those that would be missed by viewers with low vision, including actions, costumes, gestures, scene changes, facial expressions, and onscreen text. Audio description can be found in TV, movies, live theatre, museums, 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 that 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.

As of this writing, there are no known social media platforms that 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.

#### YouTube

YouTube offers several features that assist in providing closed captions for uploaded videos. These include a feature that recognizes speech within the video to generate a text transcript, a feature that synchronizes captions with 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.

#### Sites Without Captioning Support

For social media sites that 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 that 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.

### Widgets and Third-Party Sharing

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 that 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 provide insertable code to make it easier for page developers to display social content on a web page. Social sharing services offer a similar feature through widgets that give users the ability to share a 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 that does not contain any actual content. 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 convey confusing information about the frame.

Many social media sites and social sharing services generate code that 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 that 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 that 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 embedded on their web page from a social media site exposes the embedded information to assistive technologies.

## Reaching Constituents with Disabilities

By following these principles to make websites and social media accessible, people with disabilities can participate in social media. To effectively incorporate people with disabilities in outreach efforts, it is important to understand how to adjust processes to improve interaction and participation.

### Using Websites for Outreach

Each agency 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 effective communication. Since users with disabilities may access social networks using any number of applications or interfaces, avoid relying on features that are available only through the standard social media website. When special features are used, provide alternate methods to allow users with disabilities to equally participate and contribute.

Due to the many accessibility challenges that exist with social networking sites, move users away from the standard interface of social networks as much as possible. Most information posted to a social network should also be included on the agency website. Use social networks primarily to share new information with users and interact with them, then direct the user to an accessible agency website for more information. 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, link to a version of a video that includes an audio description track.

Agencies should include contact information in their profile to enable users with disabilities to reach them about 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 are effective methods for providing accessible information as the format and presentation are under greater control of the author. Updates are often easier for users to obtain because most newsletters will push the update to the user through email, and blogs can push updates 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 that contains complex formatting with certain email clients for newsletters that are sent by email.

Features that allow readers to discuss content should be provided in an accessible fashion. One major barrier that 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 it also prevents users with disabilities from participating in the discussion. A better alternative to CAPTCHA is to use an account registration system that 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. 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 in the form of accessible iPhone and Android apps that display information posted to social media channels and allow 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 can 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. Disability-related information is shared through email, social media channels, and other channels created specifically for users with disabilities. Examples of disability resources that can be engaged by agencies include Disability.gov and the Job Accommodation Network. Other District government agencies should be engaged and provided with information 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 that provide direct services to people with disabilities. These groups can pass the information on to 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: Paragraphs (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