Air Force Section 508 Playbook

Last updated 7/31/2020

Contents

[Intended Audience 4](#_Toc47102291)

[Purpose 4](#_Toc47102292)

[What is Section 508? 4](#_Toc47102293)

[Examples of ICT Include But Are Not Limited to: 5](#_Toc47102294)

[Commonly Used Products by Individuals with Disabilities: 5](#_Toc47102295)

[Universal Standards 5](#_Toc47102296)

[Application within the Air Force 6](#_Toc47102297)

[Procurement 6](#_Toc47102298)

[Stakeholders 6](#_Toc47102299)

[Section 508 Requirements for IT Acquisitions: 6](#_Toc47102300)

[Sample Contract Language: 6](#_Toc47102301)

[VPAT (Section 508 Accessibility Template) 8](#_Toc47102302)

[Website and SharePoint Manual Testing 8](#_Toc47102303)

[Testing Tools 8](#_Toc47102304)

[Analyzing VPAT Criteria and Conformance Standards with the ANDI Tool 10](#_Toc47102305)

[Auto-Playing and Auto-Updating Content: 10](#_Toc47102306)

[Keyboard Access and Focus 11](#_Toc47102307)

[Forms 13](#_Toc47102308)

[Links and Buttons 15](#_Toc47102309)

[Images 16](#_Toc47102310)

[Adjustable Time Limits 17](#_Toc47102311)

[Repetitive Content 17](#_Toc47102312)

[Content Structure 19](#_Toc47102313)

[Language 21](#_Toc47102314)

[Page Titles, Frames, and iFrames 21](#_Toc47102315)

[Sensory Characteristics and Contrast 22](#_Toc47102316)

[Tables 23](#_Toc47102317)

[Pre-Recorded Audio-Only, Video-Only, and Animations 24](#_Toc47102318)

[Synchronized Media 25](#_Toc47102319)

[Resize Text 26](#_Toc47102320)

[Multiple Ways 26](#_Toc47102321)

[**Analyzing Conformance Standards with the WAVE Tool** 27](#_Toc47102322)

[Free Automated Public-Facing Webpage Testing 27](#_Toc47102323)

[What Do All These Icons Mean? 27](#_Toc47102324)

[Did My Page Pass WAVE? Is it "WAVE Approved"? 28](#_Toc47102325)

[What Guidelines and Standards Does WAVE Use? Section 508, WCAG, or Both? 28](#_Toc47102326)

[Styles and Code 28](#_Toc47102327)

[How is the WAVE Extension Different? 28](#_Toc47102328)

[The WAVE Report is 'Broken' or Difficult to Read 29](#_Toc47102329)

[WAVE Reports an Error, but I Don't See a Red Icon 29](#_Toc47102330)

[Javascript and CSS 29](#_Toc47102331)

[Is WAVE Accessible? 29](#_Toc47102332)

[Add a WAVE Link to Your Web Page 29](#_Toc47102333)

[Use the Browser Bookmarklet 29](#_Toc47102334)

[**Website and SharePoint Reporting Metrics** 30](#_Toc47102335)

[How to Fill Out the Website Testing Worksheet: 30](#_Toc47102336)

[Part 1: Enter Testing Info 30](#_Toc47102337)

[Part 2: Enter Conformance Results 31](#_Toc47102338)

[Website and SharePoint Conformance Reporting: 33](#_Toc47102341)

[How to Enter Results: 33](#_Toc47102342)

[Creating Accessible PDFs 35](#_Toc47102343)

[Workflow for Creating Accessible PDFs 35](#_Toc47102344)

[Add other accessibility features to the PDF: 35](#_Toc47102346)

[Tag the PDF 35](#_Toc47102347)

[Watermarks and Screen Readers 36](#_Toc47102348)

[Evaluate the PDF and Repair Tagging Problems 36](#_Toc47102349)

[Create a Tagged PDF from a Web Page 37](#_Toc47102350)

[About Tags in Combined PDFs 37](#_Toc47102351)

[About Tools for Creating Accessible PDF Forms 38](#_Toc47102352)

[Authoring Applications 38](#_Toc47102353)

[Workflow for Creating Accessible PDF Forms 39](#_Toc47102354)

[Design the Form for Accessibility 39](#_Toc47102355)

[Set and Test the Tab Order of a Form 39](#_Toc47102356)

[Tool Kit 40](#_Toc47102357)

**Introduction**

# Intended Audience

Section 508 Coordinators, Website Managers, IT Portfolio Program Managers, and Procurement Officials

## Purpose

The Air Force Section 508 Playbook provides a framework to ensure U.S. Federal Government technology is accessible for individuals with disabilities. The Playbook provides guidance to the Air Force to ensure information and communication technology (ICT) is compliant with Section 508 guidelines and standards. The Playbook contains “how-to” guidance on completing the Voluntary Product Accessibility Template (VPAT)/Section 508 Template, testing websites and SharePoint sites, procurement requirements, and reporting metrics.

Numerous benefits will accrue from implementing this framework:

* Education and awareness of Section 508 requirements
* Strategic planning for increased agency Section 508 compliance and effective management of information technology
* Expanded access to Federal government electronic services by individuals with disabilities
* Increased diversity of federal workforce enabled through inclusive technology

## What is Section 508?

Section 508 of the Rehabilitation Act, as amended in 1998, requires Federal agencies to make ICT accessible to employees and members of the public who have disabilities in a comparable manner to the access experienced by employees and members of the public without disabilities.

As of January 18, 2018, Federal agencies must comply with the Revised 508 Standards, which were issued by the U.S. Access Board in January 2017. The U.S. Access Board establishes Section 508 standards in order to implement the law. These revised standards are set forth at 36 CFR Part 1194.1 and Appendices A, C and D to Part 1194 (compliance criteria). The revised Section 508 Standards apply to ICT that is "procured, developed, maintained, or used" by agencies of the Federal government. Section 508 was enacted to eliminate barriers to ICT, make opportunities available for persons with disabilities, and encourage development of technologies that will help achieve these goals. The Section 508 standards are the technical requirements and criteria used to measure conformance within this law. More information on Section 508 and the technical standards can be found at [Section508.gov](http://www.section508.gov)[.](http://www.section508.gov/)

The Section 508 law is broad in scope, applying to all information and communication technology the federal government buys, builds, maintains, and uses. Non-compliance can result in time consuming and costly lawsuits, delayed implementation of key IT investment priorities, and damage to public missions or image.

### **Examples of ICT include but are not limited to:**

* Computers and peripheral equipment
* Information kiosks and transaction machines
* Telecommunications equipment (telephones, telephone systems)
* Customer premises equipment (servers, routers)
* Multifunction office machines
* Software, applications, and websites
* Videos
* Electronic documents

### **Commonly used products by individuals with disabilities:**

* Desktop and mobile telephones and other telecommunications products that interact with users in real time
* Information kiosks and booths that provide information in public places such as Federal buildings and hospitals
* Documents that are posted to the Internet (e.g., PDF, Word, Excel, and PowerPoint)
* Multifunction machines that scan, fax, print, etc.
* Websites including content accessed from the Internet and on private networks
* Computer software and hardware including desktop systems and mobile systems such as laptops and other mobile computers

**These products are not considered ICT:**

* An air conditioning system that has a self-monitoring thermostat embedded in the unit
* Medical equipment where information technology is integral to its operation, such as x-ray machines and other diagnostic equipment

**The following are some products which may be ICT but that are not addressed by**[**Section 508 Standards**](https://training.section508testing.net/mod/url/view.php?id=504)**:**

* CDs, and DVDs (content recorded to these products must be accessible)
* Cables and power cords
* Wi-Fi, fiber optics

### **Universal Standards**

The Revised 508 Standards incorporate by reference the Web Content Accessibility Guidelines (WCAG) 2.0, a globally-recognized and technologically-neutral set of accessibility guidelines for web content.

For Section 508-covered ICT, all web and non-web content and software (such as websites, intranets, word processing documents, PDF documents, project management software, etc.) is required, with a few specific exceptions, to conform to WCAG 2.0's Level A and Level AA Success Criteria and Conformance Requirements.

By applying a single set of requirements to websites, electronic documents, and software, the revised requirements adapt the existing [Section 508 Standards](https://training.section508testing.net/mod/url/view.php?id=504) to reflect the newer multifunction technologies (e.g., smartphones that have telecommunications functions, video cameras, and computer-like data processing capabilities) and address the accessibility challenges that these technologies pose for individuals with disabilities.

## Application within the Air Force

When a new product or service is being created, it is the responsibility of everyone involved to address [Section 508 Standards](https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-ict-refresh/final-rule/text-of-the-standards-and-guidelines) during the planning, design, and development processes. The entire team (including project managers, user interface designers, programmers, developers, and possibly instructional designers) is responsible for developing a product that meets the [Section 508 Standards](https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-ict-refresh/final-rule/text-of-the-standards-and-guidelines).

***Federal agencies are subject to***[***Section 508 Standards***](https://training.section508testing.net/mod/url/view.php?id=504)***during four specific ICT lifecycle-related activities:***

* Development (Development can take many shapes such as: Software; Websites; Hardware; and Documents)
* Procurement
* Maintenance
* Use

# **Procurement**

### **Stakeholders**

Requirements Official, Contracting Officers, Program Managers, GPC Card Holders

### **Section 508 Requirements for IT Acquisitions:**

* Federal Acquisition Regulation (FAR) Subpart 39.2 (codification language).
* When you develop, procure, maintain, or use ICT, “Think 508 first!”
	+ Determine what ICT Section 508 standards apply for each procurement, and to assist with preparing requirements for market research, use the Accessibility Resources Tool (ART) tool at <https://www.section508.gov/buy/accessibility-requirements-tool>.
	+ Ensure appropriate Section 508 standards are included in requirements documents [i.e. Performance Work Statement (PWS), Statement of Work (SOW), Statement of Objectives (SOO), etc.].
	+ Generate applicable Section 508 contract language at: <https://www.section508.gov/buy/define-accessibility-criteria>
	+ Verify compliance with 508 standards before accepting deliverables, purchasing products and services, or launching websites and applications by ensuring receipt of a Voluntary Product Accessibility Template (VPAT)/(Section 508 template) from ICT vendors/developers.
	+ Ensure Section 508 is included in the vendor evaluation criteria.
	+ Maintain a dialogue regarding Section 508 with developers, contractors, and project teams, throughout the acquisition and life cycle process.

### **Sample Contract Language:**

The offeror shall support the Government in its compliance with Section 508 throughout the development and implementation of the work to be performed. Section 508 of the Rehabilitation Act of 1973, as amended (29 U.S.C. 794d) requires that when Federal agencies develop, procure, maintain, or use electronic 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 do not have 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.

Applicable standards of {insert IT system} shall be compliant with Section 508, Americans with Disabilities Act as defined in Section 508 (29 U.S.C. 794d), Technical Standards (Subpart B), Web based Intranet and Internet Information and Applications (1194.22). {insert IT system} will follow, at a minimum, the Priority 1 Checkpoints of the Web Content Accessibility Guidelines 2.0 (WCAG 2.0) (January 2018) published by the Web Accessibility Initiative of the WorldWideWeb Consortium (W3C)."

The offeror should review the following websites for additional 508 information: <https://www.section508.gov/index.php>; <https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards>

The offeror must indicate in its response package where full details of compliance to the identified standards can be found, such as vendor’s website, etc., or provide documentation using the Voluntary Product Accessibility Template (VPAT) of those applied standards during the development of the product.

The Air Force is committed to making its electronic and information technologies accessible to individuals with disabilities by meeting or exceeding the requirements of Section 508 of the Rehabilitation Act.

We require offerors to ensure all electronic information and communication technology (ICT) systems, applications, web content, reports, and deliverables be accessible to people with disabilities (per Accessibility Requirements (Section 508) and best practices (W3C WCAG 2.0).

Where not fully compliant with the technical standards, offerors should be prepared to:

* Provide clear points of contact to help users obtain alternate formats;
* Plan to fix any non-compliant items in coordination with vendor, development, or integration teams; and
* Work with project staff to develop accommodation plans, should users not be able to gain access or use the system due to non-conformance with the Section 508 technical standards.

Before final acceptance of any ICT item, including updates and replacements, if the offeror claims its products or services satisfy the applicable Revised 508 Standards specified in the statement of work, and the contracting officer determines that any furnished ICT item is not in compliance with such requirements, the contracting officer will promptly inform the offeror in writing of the noncompliance. The offeror shall, at no cost to the agency, repair or replace the non-compliant products or services within the period specified by the contracting officer.

***NOTE: Failure to consider Section 508 conformance throughout the entire development lifecycle will likely relegate 508 conformance considerations to the end of the project, when it costs the most to fix accessibility problems.***

# **VPAT (Section 508 Accessibility Template)**

An Accessibility Conformance Report (ACR) is the most common way vendors report on product conformance to the [Section 508 standards](https://training.section508testing.net/mod/url/view.php?id=504). The ACR is created using a template called the Voluntary Product Accessibility Template (VPAT), and the version of the template contains current [WCAG 2.0, 508 standards](http://www.w3.org/TR/WCAG20). [The Air Force VPAT/Section 508 Accessibility Template](https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Forms/AllItems.aspx?RootFolder=%2Fsites%2F10440%2FInfoAcc%2FSection%20508%2FSection%20508%20Accessibility%20Template&FolderCTID=0x0120009C855A49FDFEAC4DBDAF75D34D2F4BC6&View=%7B594D084C%2D6A0B%2D418F%2DA024%2DAA9DF29AA53C%7D) does not guarantee that the product is Section 508 conformant. Air Force Program Managers or Website Managers must test, verify, and validate that ICT products are Section 508 conformant.

##### **Filling Out the VPAT**

It is important to know that not every section of the VPAT will apply to ICT product or service. To determine which sections apply, review the [WCAG 2.0, 508 standards](http://www.w3.org/TR/WCAG20). Complete the VPAT using the conformance level terms defined as follows:

* **Supports:** The functionality of the product has at least one method that meets the criterion without known defects or meets with equivalent facilitation.
* **Partially Supports:** Some functionality of the product does not meet the criterion.
* **Does Not Support:** The majority of product functionality does not meet the criterion.
* **Not Applicable:** The criterion is not relevant to the product.

# **Website and SharePoint Manual Testing**

# Testing Tools

**Tools needed:** [ANDI](https://www.ssa.gov/accessibility/andi/help/install.html)**,** [Colour Contrast Analyser](https://www.dhs.gov/redirect?url=https%3A%2F%2Fgithub.com%2FThePacielloGroup%2FCCA-Win)

*\*\*How to add ANDI to the Air Force Network using Internet Explorer\*\**

*Internet Explorer*

1. *Ensure that IE's Favorites Bar is visible*
2. *Right click this link:*[*ANDI*](https://www.ssa.gov/accessibility/andi/help/install.html)*and select "Add to favorites bar". Alternatively, keyboard users running Windows can tab to this link: ANDI, press ctrl+shift+F10 (which brings up a context menu) and select "Add to favorites bar"*
3. *ANDI is now installed to IE and ready to be launched*
4. *Colour Contrast Analyser (CCA) tool can be found at:* [*https://www.dhs.gov/publication/web-accessibility-toolbar*](https://www.dhs.gov/publication/web-accessibility-toolbar)*, but access to CCA on the Air Force Network may be blocked for use depending on your location*

**Documents required to fill out:** [Website Testing Worksheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/Website%20testing%20worksheet%20sample.docx?Web=1) **and** [Website Testing Reporting Excel Sheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Forms/AllItems.aspx?RootFolder=%2Fsites%2F10440%2FInfoAcc%2FSection%20508%2FTesting%20Metrics&FolderCTID=0x0120009C855A49FDFEAC4DBDAF75D34D2F4BC6&View=%7B594D084C%2D6A0B%2D418F%2DA024%2DAA9DF29AA53C%7D&InitialTabId=Ribbon%2ERead&VisibilityContext=WSSTabPersistence)

**Recommended training:** [DHS Trusted Tester](https://training.section508testing.net/)

***\*NOTE: Upon completion and certification as a DHS Trusted Tester, submit your name and certificate to the SAF/CNZA Org Box:*** ***usaf.pentagon.saf-cn.mbx.af-section-508@mail.mil***

# **Analyzing VPAT Criteria and Conformance Standards with the ANDI Tool**

# Auto-Playing and Auto-Updating Content:

**Purpose**: This topic covers how to test content that plays or updates automatically and verify that there is a method to pause, stop, hide, or control automatic content and that the page provides notification of each automatic change of content. This ensures that all users can access all page content without disruption of AT or distracting content playing on the page.

**Tools:**

* Manual inspection of the page.
* ANDI: structures > live regions.
* Any applicable ANDI modules.

**Identifying content**

* 1.4.2-audio-control: Identify all audio content that plays automatically for more than 3 seconds.
* 2.2.2-blinking-moving-scrolling: Identify all visual content that automatically moves, blinks, or scrolls for more than 5 seconds and is presented along with other page content. EXCLUDE all content where the movement, blinking, or scrolling of the content is essential.
* 2.2.2-auto-updating: Identify all content that automatically updates and is not the only content on the page.
* 4.1.2-change-notify-auto: Identify all content that changes automatically on the page as part of an auto-update.

**Making test decisions:**

* 1.4.2-audio-control: Verify there is a mechanism within the first three elements a user would encounter that can pause/stop the audio or control the volume of the auto-playing audio and the mechanism PASSES all applicable Test Conditions.
* 2.2.2-blinking-moving-scrolling and 2.2.2-auto-updating: Verify there is a mechanism within the first three elements that the user would encounter OR within three elements before/after the moving/blinking/scrolling/auto-updating content that can pause/stop/hide content and the mechanism PASSES all applicable Test Conditions. (2.2.2-auto-updating also includes the ability to control the frequency of the updates.)
* 4.1.2-change-notify-auto: The page provides notification of each automatic update/change in content via one of the following: (1) a keyboard-accessible dialog, or (2) the page moves focus to the content that has changed, and the content that has changed provides sufficient description about the change, or (3) the content that has changed is contained in an ARIA live region.

# Keyboard Access and Focus

**Purpose:** All interactive content must be accessible to keyboard-only users and provide focus to orient them.

* Keyboard: The purpose of the Keyboard Access tests is to ensure that all functionality is available to keyboard-only users. This test also ensures that individual keystrokes for keyboard users do not require specific timings for activation of functionality and that a keyboard user does not get trapped on an element or stuck within a section of a web page. This ensures that the web page is fully operable by keyboard-only users.
* Focus: The purpose of the Focus tests is to ensure that every keyboard-accessible component has a visible indication when it has focus and that each web page can be navigated sequentially so that meaning and operability are preserved. It also verifies that when components receive keyboard focus, an unexpected change of context does not occur. Predictable focus allows keyboard users to easily operate and understand page content.

**Tools:**

* Manual inspection of the page.
* Manual inspection of the page using only a pointing device.
* Manual inspection of the page using only the keyboard only.

**Identifying content:**

* Use the mouse or other pointing device to determine available functions provided by interactive elements (including drop-down menus, form fields, revealing/hiding content, tooltips, AND all interactive interface components).
* Use the keyboard to navigate to keyboard-accessible interface components (including drop-down menus, form fields, revealing/hiding content, tooltips, AND interactive interface components). These tests ONLY apply to components that are keyboard accessible. If the keyboard cannot access the component, it is NOT TESTED under 2.1.2-no-keyboard-trap or any Focus test.

**Making test decisions:**

* Keyboard: Interactive content functionality, including title attributes, that are essential to understanding the page must be accessible using the keyboard alone (2.1.1-keyboard-access) without requiring any specific timing of the keystrokes (2.1.1-no-keystroke-timing). Further, keyboard users must be able to navigate away from each keyboard-accessible element and section of the page using ICTher standard navigation keys or documented custom keystrokes (2.1.2-no-keyboard-trap).
* Focus: When using the keyboard, there must be a visible indication of focus (2.4.7-focus-visible) indicating where you are, and the focus must not initiate an unexpected change of context (3.2.1-on-focus). Further, the meaning and operability of the page must be preserved as focus moves through the page (2.4.3-focus-ordermeaning). Finally, only one additional keystroke is allowed to move to revealed content (2.4.3-focus-orderreveal) and back to a logical sequence of focus order before the content was revealed (2.4.3-focus-order-return).

**Other considerations**

Below are some important considerations to successfully perform Keyboard Access and Focus tests and for students to successfully pass the certification exam.

A mouse or other pointing device is only used to:

* Identify items to be tested.
* Resume testing after a keyboard trap is noted.

Do NOT use a mouse or other pointing device to perform any of these tests. All tests must be performed using the keyboard ONLY. For example:

* Using a mouse to escape a keyboard trap would not pass the test.
* If there is visible focus when hovering over an element with a mouse, this would NOT pass a 2.4.7-focus-visible test. You must evaluate whether the item receives visible focus using the keyboard.

 If an element CANNOT be reached by keyboard, it is NOT TESTED under any Focus test.

For 3.2.1-on-focus, ensure you understand the difference between changes of context versus changes of content. You are specifically looking to ensure that when an element receives focus, there is not a change of context. Changes in context include unexpected outcomes such as a new window being launched, an action being performed, or focus moving to another interface component. Changes in context are always expected to be initiated by the user and should never occur on focus. On the other hand, changes in content ARE NOT ALWAYS changes in context. Changes in content could be expanding an outline, revealing/hiding content, or accessing a dynamic menu. These are not unexpected and may not change the meaning of the page itself.

For 2.4.3-focus-order-return, testers must pay attention to where they are on the page when selecting a component that reveals hidden content. When exiting the revealed content, focus must be returned to a logical sequence of focus order.

For 2.1.1- keyboard-access, you do not need to evaluate every instance where a title attribute is used on the page. Determine instances where the title attribute provided essential information to understand the page content or complete an activity. You need to be able to identify a modal dialog box. These are boxes that must be closed before users can interact with the page that originated the modal dialog box. “Save As” is an example of a modal dialog box. Users must indicate where they want to save and what name to use OR they must cancel the action before the box will close. No other actions can be performed until the Save As box closes. By definition, keyboard focus should remain within the modal dialog box while it is open.

# Forms

**Purpose:** This topic covers how to verify that forms and form notifications provide complete on-screen instructions/labels and correct programmatic association so all users can interact with the forms in a predictable way. This test topic also covers changes to web page content that occur as the result of interacting with forms.

**Tools:**

* Keyboard.
* ANDI: focusable element, tables, structure.
* Manual page inspection.

**Identifying content:**

* 3.3.2-label-provided, 2.4.6-label-descriptive, 1.3.1-programmatic-label, and 3.2.2-on-input: Using ANDI, find all form elements and all instructions and cues (textual and graphical) that are related to form components/controls, e.g., groupings, order of completion, special conditions, qualifiers, format instructions. EXCLUDE disabled input elements.
* 4.1.2-change-notify-form: Find all form elements that change page content due to form interaction. This test will apply only if page content changes as a result of interacting with form elements; if no content changes this test DOES NOT APPLY.
* 3.3.1-error-identification: Use ANDI to identify any enabled form elements on the page. Find all instructions and cues (textual and graphical) that are related to form components/controls, e.g., groupings, order of completion, special conditions, qualifiers, format instructions. Intentionally enter values and/or make selections that violate format and/or other form instructions and only test form fields that have automatic error detection.
* 3.3.3-error-suggestion: Find all form fields that have automatic error detection. Do not test if any of the following apply to the page/form elements that have automatic error detection:
	+ Based on the type of input required, suggestions for correction cannot be provided because they are not knowable.
	+ Providing information about how to correct the error would jeopardize the security or purpose of the content, e.g., details about an incorrect password.
* 3.3.4-error-prevention: Find all content that submits user form entries that result in or cause legal commitments or financial transactions, submits user form entries that modify or delete user controllable data in a data storage system, and/or submits user test responses.

**Making test decisions:**

* 3.3.2-label-provided: Check that each form field has a label or instructions.
* 2.4.6-label-descriptive: Review the labels and instructions for each form field identified and validate that each form label/instruction is sufficiently clear and descriptive so that users know what input data is expected.
* 1.3.1-programmatic-label: Using ANDI: focusable elements, verify that the ANDI Output includes all relevant instructions and cues on the page for the form element, including when fields are required. If ANDI: focusable elements does not provide sufficient information, review other programmatic associations, such as table headings (row/column) or location in a hierarchical list structure, to determine whether they provide or contribute to the form field’s description, cues, or instructions.
* 3.2.2-on-input: Use the keyboard to navigate to form elements and complete the form elements (e.g., type text in to the input field, select a radio button). Navigate away from the completed form field and check for any instances of an unexpected change of context. Determine whether there are unexpected changes in context (e.g., changes of user agent, viewport, focus, or change of page meaning due to such things as automatic submission).
* 4.1.2-change-notify-form: Determine whether the user is made aware of any changes to page content initiated by interaction with form elements in any of the following ways: accessible name/description of triggering element, keyboard-accessible dialog alert, notification of focus shift to changed content, or use of a live region.
* 3.3.1-error-identification: Intentionally violate formatting and required form elements and submit the form and/or move to the next page to identify if an input error is automatically detected. Determine whether the item that is in error is identified in text and described to the user in text.
* 3.3.3-error-suggestion: Determine whether the error description contains adequate information for the user to know what is required to fix the error and/or offers suggestions for correction.
* 3.3.4-error-prevention: Complete the required form fields with intentional errors and submit the content. Determine whether the user can reverse the submission, is offered the option to review, confirm, and correct information before finalizing the submission, OR the page checks data for input errors and allows the user an opportunity to correct any errors.

**Other considerations:**

Remember: Test 5.A for 3.2.2-label-provided is verifying that a label or instructions are provided for each form field. In 5.B for 2.4.6-label-descriptive you verify if the label or instructions are sufficiently descriptive for each form element.

To successfully pass the certification exam for 3.2.2-on-input, you are looking for an unexpected change of context. A change of context is a major change in content that, if made without user awareness, can disorient users. Several change-in-context examples are:

* Forms submitted automatically when exiting the field.
* Forms submitted automatically when exiting the last field in a form.
* New window launched when changing a radio button selection.
* Focus is changed to another component when a select list item is selected.
* Anything that gives the appearance of moving the user to a new page.

Changes of context ARE ALLOWED IF the change is not unexpected (Either the user is provided advanced notification of the change or the control is clearly intended to change context).

Be sure to change the form element value and then navigate away from the element to verify that there is no unexpected change of context.

# Links and Buttons

**Purpose:** Link/Button Purpose: The purpose of this test is to determine whether a link or button provides sufficient description of the link or button’s purpose. To decide whether to follow a link or activate a button, users need to understand what action will occur once they click the link or button. Developers can describe the purpose of the link or button directly in the link or button text by programmatically associating other descriptive text with the link or button, and/or by providing contextual cues that can be determined programmatically. This test describes the steps necessary to evaluate the various methods that a developer may use to describe a link or button.

Link/Button-Related Changes: The purpose of this test is to determine whether the web page provides sufficient notification and description about any changes to content on the web page that occur as a result of interaction with a link or button. Developers may use the link or button text and other associated text identified in the previous test to provide notification of link/button-related changes. Developers may also use other techniques to provide notification of link/button-related changes. This test describes the steps necessary to evaluate the various methods a developer may use to notify the user about link or button-related changes.

**Tools:**

* ANDI: links/buttons > links.
* ANDI: links/buttons > buttons.
* ANDI: tables.
* ANDI: structures > lists.
* ANDI: structures > live regions

**Identifying content:** Use ANDI: links/buttons to identify all links and buttons.

* Identify any content changes to the page that occur as the result of interaction with links or buttons. Changes in content may include changes to other links and buttons, textual information, page structure, images, graphs, color schemes, etc.

**Making test decisions:**

* 2.4.4-link-purpose: Evaluate the ANDI Output for each link to determine whether the link or button’s accessible name and description provide adequate description of its purpose.
* 4.1.2-change-notify-links: When a user activates a link or button that results in web page content changes, evaluate whether any one of the following is true:
	+ The button or link provides adequate notification and description of the change.
	+ The page provides notification of the change via a keyboard-accessible dialog.
	+ The page moves focus to the content change, and the content itself provides adequate notification and description of the change.
	+ The content that changed is contained in an ARIA live region.

**Other considerations:**

Any changes to links or buttons that occur automatically or as a result of interaction with the page should be included in this test.

To successfully evaluate programmatically determined context, it may be necessary to use other ANDI modules and features aside from the links/buttons module (e.g., tables, structure > lists). To successfully conduct this specific set of tests, you will also need to be familiar with the other Test Conditions in this test process.

# Images

**Purpose:** The purpose of this topic is to verify that:

* Meaningful images have an appropriate equivalent so that all users can access the information contained within the graphic.
* Decorative images do not have an alternative text description (as indicated by either a blank ANDI Output or inserted as a background image).
* CAPTCHA images provide alternative methods for users that do not only rely on vision or hearing.
* Presentation of images of text is essential or the text is customizable.

**Tools:**

* ANDI: graphics/images and hide/find background.
* Manual inspection.

**Identifying content:**

* For 1.1.1-meaningful-image-name: Use ANDI: graphics/images to find all images. Start with the first image outlined by ANDI: graphics/images and find all meaningful images. An image that is on the page but not detected by ANDI should NOT be included in this test.
* For 1.1.1-decorative-image: Use ANDI: graphics/images to find all images. Start with the first image outlined by ANDI: graphics/images and find all decorative images. An image that is on the page but not detected by ANDI should NOT be included in this test.
* For 1.1.1-decorative-background-image: Use the ANDI: graphics/images to find CSS background images. If the “find background” and “hide background” buttons are available, background images are on the page.
* For 1.1.1-captcha-alternative: Find all CAPTCHA images.
* For 1.4.5-image-of-text: Use ANDI: graphics/images to find all images of text, excluding text that is part of a picture that contains significant other visual content such as graphs, screenshots, and diagrams, which visually convey important information more than just text.

**Making test decisions:**

* For 1.1.1-meaningful-image-name: Determine if ANDI Output contains an equivalent description for the meaningful image and/or refers to a description in the page content.
* For 1.1.1-decorative-image: ANDI Output for a decorative image is blank.
* For 1.1.1-decorative-background-image: The background image is decorative OR the meaning of the background image is also available without the background image.
* For 1.1.1-captcha-alternative: Determine if the CAPTCHA has a format for users without vision AND a format for users without hearing.
* For 1.4.5-image-of-text: Determine if the image of text cannot be replaced with text (e.g., the image is a logotype, type sample, branding, images of specific fonts that are not widely supported) OR the image of text can be visually customized for font, size, color, and background by controls provided on the web page.

**Other considerations:**

IMPORTANT: To successfully pass the certification exam for this item, you need to understand the difference between meaningful and decorative images:

* A meaningful image is one where information is being conveyed and, if removed, there would be a loss of content or context.
* A decorative image is an image that is serving only an aesthetic purpose (e.g., borders, spacers, and corners), is invisible/not presented to the user, is part of a link to improve appearance or increase clickable area, used for visual interest only, or described by surrounding text.

# Adjustable Time Limits

**Purpose:** The purpose of this test topic is to ensure that users with disabilities are given adequate time to interact with web page content. It may be difficult for some users to perform required actions before a time limit occurs if there is no way to turn off, adjust, or extend the time limit.

**Tools:** Manual inspection of the page.

**Identifying content**: Identify any instances of content time limits, excluding real-time exceptions, essential exceptions, and 20-hour exceptions. Time limits could be identified by:

* Inspecting the system or site documentation.
* Looking for indicators of time limits on the page where the time limit occurs, including:
	+ Text description on the page.
	+ Pop-ups, messages, or warning indicators on the page.
	+ Allowing the page to be idle for an extended period of time to prompt a time-out notification or other indication that a time limit has occurred.

**Making test decisions:** Determine whether the web page provides a way to turn off, adjust, or extend the time limit.

# Repetitive Content

**Purpose:** When there are repetitive blocks of content on multiple pages in a website, there must be a keyboard-accessible method for users to bypass each block of repetitive content. Repetitive content must appear in the same relative order on the page and with respect to other repetitive content. Furthermore, when the same function appears across the website, it must be consistently labeled and consistently identified. Content must appear on multiple pages for it to be considered repetitive and for these tests to apply. The results from the tests in this topic are used to determine if the following WCAG SC are met:

* 2.4.1-bypass-function
* 3.2.3-consistent-navigation
* 3.2.4-consistent-identification

**Tools:**

* ANDI: Focusable Elements > “tab order” enabled.
* ANDI: links/buttons > “Show Links List” button > skip links (if available).
* ANDI: structures > “reading order” enabled.
* Manual inspection.
* Keyboard.

**Identifying content:**

* 2.4.1-bypass-function: Identify block(s) of content that are repeated on other pages within the site. This Test Condition DOES NOT APPLY to a web page that does not contain blocks of content that are repeated on other web pages. If there is only one web page that is not part of a larger website, this test would not apply.
* 3.2.3-consistent-navigation: Identify all navigational elements that are repeated on other pages within the website. This Test Condition DOES NOT APPLY to a web page that does not contain navigational elements that are repeated on other web pages. If there is only one web page that is not part of a larger website, this test would not apply.
* 3.2.4-consistent-identification: Identify components that are repeated on other pages and have the same functionality within a set of web pages. This Test Condition DOES NOT APPLY if there are no components that have the same functionality within a set of web pages. If there is only one web page that is not part of a larger website, this test would not apply.

**Making test decisions:**

* 2.4.1-bypass-function: Using ANDI: focusable elements and/or ANDI: links/buttons > “Show Links List” button > skip links (if available), identify all bypass options. Then, use the keyboard to verify the skip function works and moves the user past repetitive blocks of content. If there is content that is NOT repetitive between two repetitive blocks, then each block of repetitive content needs a separate bypass function. It may not be considered one repetitive block.
* 3.2.3-consistent-navigation: Using ANDI: structures > “reading order” and manual inspection, verify that each repeated component occurs in the same relative location on the web page with regard to the other repeated components on each web page where it appears.
* 3.2.4-consistent-identification: The accessible name and description in ANDI Output is consistent for components that perform the same function on other pages of the website.

**Other considerations:**

To recap, Topic 9 is looking to ensure the navigational elements are in the same relative order. Same relative order is defined as the same position relative to: (1) the placement compared to other web pages, and (2) to the order of the navigational elements themselves.

**IMPORTANT:** For 3.2.3-consistent-navigation, it is important to understand that the elements do not have to be interactive. They can be static, repeated text, such as a header or footer, or a definition box that has a consistent placement on the page to provide a consistent layout. Navigational elements are broad to encompass any element or component that may help users orient themselves to and use a web page. These elements must repeat across web pages in a website for this test.

To successfully pass the certification exam for 3.2.4-consistent-identification, you need to understand that the test is looking for the same functionality, meaning the component produces the same result.

An important caveat to note: This test is not looking to make sure that the same graphic has the same ANDI Output every time it appears (which was a requirement for the prior Trusted Tester processes). Instead, this test first looks at the functionality of the component and then checks that each time that functionality appears, the labeling is consistent. The same graphic could be repeated across the website but provide different functionality. If labeled consistently, this would pass 3.2.4-consistent identification.

It is also important to understand that consistent text alternatives for interface elements that perform the same function may not always be exactly “identical,” such as an arrow labeled “Go to Page 4;” the same arrow image on the next page should then state “Go to Page 5.”

# Content Structure

**Purpose:** The purpose of this topic is to validate that the visual content structure elements of headings and lists are also programmatically determinable. If a visual heading or list is used, it must also be coded properly. This ensures that AT users can understand page structure and the order of content.

**Tools:**

* Manual inspection of the page.
* ANDI: structures > headings > view outline.
* ANDI: structures > lists.

**Identifying content:**

* For Heading Purpose (2.4.6-heading-purpose) and Lists (1.3.1-list-type), the tests require the tester to identify visually apparent headings and lists.
* Both visually apparent headings and programmatic association are used to identify content for Determinable Heading (1.3.1-heading-determinable).
* Programmatic headings are used to identify content for Heading Level (1.3.1-heading-level).
* Pay close attention to when visually apparent information is being validated versus when programmatic association is being validated.

**Making test decisions:**

* Heading Purpose (2.4.6-heading-purpose): No ANDI tools are used. Visually identify the heading and validate that the heading describes the topic or purpose of its content. If a heading is provided, it must be descriptive.
* Determinable Heading (1.3.1-heading-determinable): Select ANDI: structures and check the ANDI Output for each visually apparent heading to ensure all visual headings are programmatically defined. You must also review each heading identified by ANDI to verify it is a visually apparent heading.
* Heading Level (1.3.1-heading-level): Launch ANDI: structures and select the “view outline” button to display the structure outline. Verify that every programmatically identified heading level logically matches the visual heading structure on the page and that ANDI does not show a heading level conflict alert.
* Lists (1.3.1-list-type): Launch ANDI: structures and select the “lists” button. Validate that visually apparent lists are programmatically identified according to their type and all programmatic list relationships, including nesting and hierarchies, are consistent with the list relationships presented visually.

**Other considerations:**

Ensure than when testing, ALL Evaluate Results are considered before passing or failing a Test Condition. For Topic 10, if ONE Evaluate Result is FALSE, the Test Condition is FALSE. Further, if only a portion of a test DOES NOT APPLY (such as a description list not being present on the web page), you do not consider that part of the test step, and instead use results from the other test steps.

Here are some tips for other tests in Content Structure:

* Test Condition 10.A, 2.4.6-heading-purpose, is a slightly different test. No test tools are used; you are simply ensuring that the visual heading accurately describes the purpose of the content. It does not matter if the heading is actually coded as a heading for Test Condition 10.A—if it visually looks like a heading, it is checked here.
* For Test Condition 10.C, 1.3.1-heading-level, the heading level structure should be logical. It does not ALWAYS have to be in sequence, as long as the structure outline shows heading levels that logically match the visual presentation. To recap:

1. On pages that have only one heading, that heading can have any heading level, as the page’s heading level structure is defined by that one heading.

2. The most important heading(s) should have the highest priority level. For example, heading level 1 is a higher level than heading level 2, which is higher than heading level 3.

3. Headings with an equal or higher level start a new section; headings with a lower level are subsections that are part of the higher-leveled section.

4. A heading level 1:

* Is not required.
* Can be used more than once on a page.
* Is not required to match the page title.

5. The level of headings may not always be in sequence but may be valid as it relates to the visual structure/importance communicated by visible headings on the page. For example:

* An <h2> heading may be used for a navigation structure that precedes an <h1> title on a page.
* It is acceptable to have <h3> and then <h5> without an <h4> in between.
* When performing List testing, 1.3.1-list-type, it is very helpful to understand the HTML structure of lists. Unordered lists, such as those that use bullets, use <ul> and <li>. Ordered lists (those that are numbered or use another specified order) use <ol> and <li>. Descriptive lists (those used to define terms) use <dl>, <dt> , and <dd>.

# Language

**Purpose:** Every page must properly identify all human language used on the page so that assistive technology can properly render the spoken content. The results from the tests in this topic are used to determine if the following WCAG Success Criteria (SC) are met:

* 3.1.1 Language of Page
* 3.1.2 Language of Parts

**Tool:** ANDI: structures

**Identifying Content:** Languages used on the page need to be specified using the correct two or three letter code (following IANA standards).

* 3.1.1-page-language-defined always applies. If no language is visible on the page, review alternative text associated with images, transcripts, or closed captioning of multimedia to determine the language.
* 3.1.2-part-language-defined only applies when more than one language is on the page. Words that are gibberish, a proper name, a technical term, or a commonly understood word or phrase are not tested for language of parts.

**Making test decisions:** The language attribute result in ANDI must correctly display the number of human languages on the page. The HTML elements associated with the language attribute must match the portion containing the language. Any languages on the page must be associated with the correct IANA code for the language being used.

# Page Titles, Frames, and iFrames

**Purpose:** The purpose of this test is to programmatically define a descriptive page title, frame title, and/or iframe name. These titles/names allow users to quickly and easily identify whether the information contained within the page/frame/iframe is relevant to them and choose whether or not to interact with the content. These titles/names also provide users the ability to differentiate content.

* The results from the tests in this topic are used to determine if the following WCAG SC are met:
	+ 2.4.2-page-title-defined
	+ 2.4.2-page-title-purpose
	+ 4.1.2-frame-title
	+ 4.1.2-iframe-name

**Tools:**

* ANDI: structures > “Accessibility Alerts” and ANDI: structures > more details > page title.
* Manual inspection of the page for content/purpose.
* ANDI: iframes.

**Identifying content:**

* 2.4.2-page-title-defined and 2.4.2-page-title-purpose: These tests can be performed concurrently and apply to all web pages.
* 4.1.2-frame-title: Launch ANDI to determine if this test applies. If a frame is used, a message box appears as notification that frames have been detected. If there are no frames, ANDI provides no notification, and this test Does Not Apply.
* 4.1.2-iframe-name: Launch ANDI: iframes to navigate to and highlight iframes on the page. When you select the “Next Element” button, you should test any iframes that do not have a tab index that is a negative value. This includes iframes with a value of 0 or 1 or no tab index listed in Accessibility Components. This test does not apply if:
	+ There is not an option for the iframes module when you launch ANDI; there are no iframes on the web page.
		- The iframe module appears, but when you select the “Next Element” button a negative tab index (such as -1) appears; this means that the iframe is not in the tab order.

**Making test decisions:** Below is a summary of how to make test decisions

* 2.4.2-page-title-defined: Launch ANDI: structures. Check the alerts in ANDI’s “Accessibility Alerts” section to determine whether ANDI displays any of the following invalid HTML Alerts:
	+ “Page has no <title>”
	+ “Page <title> cannot be empty”
* 2.4.2-page-title-purpose: Launch ANDI: structures, then select “More Details,” then “Page Title.” Evaluate the purpose and content of the web page then check that the page title in the dialog box is a meaningful representation or indication of page content. Check that the page title accurately identifies the contents or purpose of the web page. If the web page is part of a set of web pages, check that the page title accurately distinguishes the web page from other pages in the website.
* 4.1.2-frame-title: Launch ANDI. If a frame is used, ANDI will provide a notification that frames have been detected. Select “Cancel” on the message to display the links to any frames. Select each link to access and test each frame. Review the frame to understand its content. Use the browser’s “Back” button to return to the page being tested and launch ANDI again. In ANDI, check each frame’s title attribute for a meaningful description of the content.
* 4.1.2-iframe-name: Launch ANDI: iframes and review the ANDI Output for each iframe that has a tab index that is not a negative value or displays no tab index listed under “Accessibility Components.” Determine whether the accessible name and description accurately describe the content of each <iframe>.

**Other considerations:**

IMPORTANT: To successfully pass the certification exam for this item, pay attention to the tab index of all iframes.

# Sensory Characteristics and Contrast

**Purpose:** The tests for Sensory Characteristics and Contrast determine if content relies solely on a particular sense to be perceivable. These tests include ascertaining if color alone is used to convey meaning (1.4.1-color-meaning), whether sensory characteristics alone are used in meaningful content or instructions (1.3.3-sensory-info), and ensuring sufficient contrast is used for text and its background (1.4.3-contrast).

The results from the tests in this topic are used to determine if the following WCAG SC are met:

* 1.4.1 Use of Color
* 1.3.3 Sensory Characteristics
* 1.4.3 Contrast (minimum)

**Tools:** The following tools are used to test this topic:

* Manual inspection of the page.
* Inspection for auditory cues.
* ANDI: color contrast.
* Colour Contrast Analyser (CCA)

**Identifying content:** Manual inspection of the page is required to determine where instructions or content rely on sensory information such as the use of color, shape, size, position, appearance, or sound. Where these are used, another method must also be provided for users who cannot perceive the sensory details.

* Testers must examine the use of sensory information and color and should refrain from testing certain instances where they are found.
* There are uses of color that are not tested for 1.4.1-color-meaning:
	+ Uses of color for decorative purposes only
* There are uses of sensory information that are not tested for 1.3.3-sensory-info:
	+ Shapes and visual details are for decorative purposes only
	+ Sounds that convey no content meaning or instructions
* Text contrast should not be tested for 1.4.3-contrast when the text is:
	+ In logotypes: logo or brand name
	+ For inactive (disabled) user interface components
	+ For purely decorative purposes
	+ Part of an image with significant other visual content

**Making test decisions:** The tester must carefully examine how sensory information is provided to determine if it is conveying information important to the content or instructions. In some instances this can be a subjective determination. Much of this testing requires manual inspection and evaluation in context with the rest of the page content.

**Other considerations:**

IMPORTANT: To successfully pass the certification exam for this topic, you need to understand the difference between meaningful and decorative uses of sensory content. Furthermore, rather than memorizing the instances where testing does not apply, you will have greater success in testing if you understand the rationale as to when sensory and color information would not need to meet the requirement.

# Tables

**Purpose:**

* Tables Purpose: When information, structures, and relationships are conveyed through presentation, a programmatic association is also needed so that information important for comprehension is perceivable to all users, including those using AT. When data tables are used, there should be appropriate programmatic association between data cells and table headers so that AT users can understand the table. When a table structure is used simply for layout purposes, data table structure elements cannot be used.

**Tools:**

* ANDI: structures > reading order.
* ANDI: tables.

**Identifying content:**

* Identify all layout and data tables (including images of data tables). Data tables are those that require header cells to make sense; if the ANDI reading order makes sense, the table is not a data table.

**Making test decisions:**

* 1.3.1-table-identification: Use ANDI: tables to make sure it is possible to navigate to each data table using the ANDI Analyze Previous/Next Table buttons.
	+ The data table DOES NOT have an ARIA role=”presentation” assigned.
	+ The data table DOES NOT have any ANDI Table alerts for incorrect use of ARIA table attributes.
* 1.3.1-cell-header-association: Use ANDI: tables to inspect each data cell and corresponding ANDI Output.
	+ Verify that the data table appropriately identifies all header relationships for each data cell.
* 1.3.1-layout-table-structure:Use ANDI: tables to inspect each table and table cells (if necessary) to validate that the layout table does not programmatically define table header structure:
	+ i.e., ANDI: tables cannot detect the table, has role=“presentation,” or
	+ there are no ARIA table attributes and table header structure elements and attributes.

**Other considerations:**

IMPORTANT: To successfully pass the certification exam for this item, you need to understand the difference between layout and data tables. Data tables are tables where information in a cell requires a row or column header to adequately describe the cell’s content. Layout tables are those that are used for placement of components on the page for visual aesthetic ONLY. Also, pay close attention to which Test Condition you are testing and the specific results you are looking for in evaluating each Test Condition against ANDI’s output.

# Pre-Recorded Audio-Only, Video-Only, and Animations

**Purpose:** The purpose of this lesson is to ensure that information presented in prerecorded audio-only is also presented in an equivalent text-based alternative and that prerecorded video-only content is provided in an equivalent alternative (Either a text-based alternative or an audio alternative).

**Tools:** Manual inspection of the page.

**Identifying content:**

* For 1.2.1-audio-transcript-text, identify all prerecorded audio-only content. EXCLUDE any audio-only content that is clearly labeled as a media alternative for text or consists of short sounds used to notify the user, such as confirmation beeps and error notifications.
* For 1.2.1-video-alternative-equivalent, identify all prerecorded video-only content. In a video only presentation, information can be presented in a variety of ways, including animation, text or graphics, the setting and background, the actions and expressions of individuals, animals, etc. EXCLUDE any video-only intended as a media alternative for text if it is clearly labeled as such.

**Note:** If media has both audio and video, use the synchronized media tests in Section 17.

**Making test decisions:**

* For 1.2.1-audio-transcript-text, verify that a text-based transcript is provided for all audio-only content AND the transcript is an accurate and complete representation of the audio-only content.
* For 1.2.1-video-alternative-equivalent, verify that a text or audio alternative is provided for all video-only content AND the text or audio alternative is an accurate and complete representation of the video-only content.

**Other considerations:**

Remember:

* For 1.2.1-audio-transcript-text, only test elements that play only audio (the element has no video or visual elements).
* For 1.2.1-video- alternative-equivalent, only test elements that only play video (the element has no audio or sound).
* For any media that plays BOTH video (visual elements) and audio (sound), test under Section 17

# Synchronized Media

**Purpose:** The purpose of this topic is to ensure that all users have an equivalent access to visual and auditory components of synchronized media. This is achieved by providing (1) synchronized captions of the audio track for individuals without hearing or who have limited hearing, and (2) a soundtrack describing information provided only visually. For live presentations, accurate captions must be provided. The purpose of this topic is also to ensure the media player used to present synchronized media must provide controls for captioning and audio description at the same level as volume/program selection controls.

**Tools:** Manual inspection of the page.

**Identifying content:**

* Identify any prerecorded synchronized media content (1.2.2-captions-equivalent and 1.2.5-audiodescription-equivalent).
* For 1.2.4-captions-live-equivalent, identify any live synchronized media content (EXCLUDE two-way multimedia calls between two or more individuals through web applications).
* For 503.4-caption-description-controls, identify any media player used to display synchronized video and audio content.
	+ Identify any media player with program selection control(s) (503.4.2-description-control). This control allows the user to choose what presentation, or portion of a longer presentation, to play. This is similar to a Table of Contents option. It does not include: play, pause, fast forward, rewind, etc.
	+ Identify any media player with volume adjustment control(s) (503.4.1-caption-control).

**Making test decisions:**

* The multimedia provides accurate captions for audio content (1.2.2-captions-equivalent) and an equivalent soundtrack for the video content (1.2.5-audio-description-equivalent).
* The live multimedia provides accurate captions for the audio content (1.2.4-captions-live-equivalent).
* The media player provides user controls for closed captions AND audio descriptions
* The media player provides user controls for (503.4-caption-description-controls):
	+ Captions at the same menu level as volume or program selection controls (503.4.1 caption-control)
	+ Audio description at the same menu level as program selection controls or volume (503.4.2- description-control).

**Other considerations:**

Remember, for 503.4-caption-description-controls, 503.4.1-caption-control, and 503.4.2-description-control you are testing the media player itself. Further, 503.4.2-description-control only applies if there is a program selection control; a program selection control allows the user to select which presentation or portion of a presentation to play.

# Resize Text

**Purpose:** The purpose of this test is to ensure that there is a non-AT reliant mechanism that allows users to enlarge text to at least 200% of its original size without any loss of page content or functionality.

**Tools:**

* Zoom mechanism (browser or other mechanism on the web page).
* Manual inspection of the page.

**Identifying content:** All text on a page, EXCLUDING captions for synchronized media and images of text.

**Making test decisions:** Use the built-in browser zoom functions to resize the text to at least 200%; if that method does not work, determine if another non-AT mechanism to resize page content to 200% of its original size is available and enable it. Check that all of the following are true: The non-AT-reliant mechanism allows the user to resize text to at least 200% of its original size.

* + When text is enlarged, it is not clipped, truncated, or obscured.
	+ When text is enlarged, all functionality and content is still available.

**Other considerations:**

IMPORTANT: Remember, this test includes transcripts, modal dialog boxes or other linked documents, but EXCLUDES captions for synchronized media and images of text.

# Multiple Ways

**Purpose:** The purpose of this test is to ensure users have at least two methods to locate content to allow them to choose a method that is easier and more comprehensible to them.

**Tools:** No specific test tool is used to test; this requirement uses manual testing.

**Identifying Content:** When identifying content to test for multiple ways to navigate to content, there must be more than one web page in the website. Any web page in the site that is accessed as a result of or step in a process, such as a shipping confirmation page, is excluded from this test.

**Making test decisions:** Websites with multiple pages must have at least 2 working methods to navigate to content. This may be, but is not limited to, any combination of the following navigation methods: site search, table of contents, navigation menus or dropdowns, navigation trees, and/or links between pages.

**Analyzing Conformance Standards with the WAVE Tool**

# **Free Automated Public-Facing Webpage Testing**

**Tools Needed:** Wave by WebAIM

[WAVE](https://wave.webaim.org/) is suite of tools to help web you make your web content more accessible. WAVE cannot tell you if your web content is accessible. Only a human can determine true accessibility. But, WAVE can help you, as a human, evaluate the accessibility of your web content.



## What Do All These Icons Mean?

WAVE will present your page with embedded icons and indicators. Each icon, box, and piece of information added by WAVE presents some information about the accessibility of your page. While WAVE is most effective when used by someone knowledgeable about web accessibility, individuals who are not web accessibility experts can also benefit from WAVE.

The WAVE sidebar indicates if WAVE has detected any errors or not. The absence of errors DOES NOT mean your page is accessible or compliant. RED icons indicate accessibility errors that need to be fixed. GREEN icons indicate accessibility features - things that probably improve accessibility (though these should be verified). The other icons and indicators, particularly the yellow alert icons, highlight other elements that you should look at. WAVE brings the underlying accessibility information of a page to the forefront so it can be easily evaluated in context.

The goal should not be to get rid of all icons, except for the errors which may represent an end user issue, but must be manually assessed. Other icons are presented to facilitate human analysis of accessibility and structure of the page.

You can view a brief overview of what each icon means by clicking it and viewing its documentation or by accessing the Reference panel in the WAVE sidebar. You can view details on all of the WAVE icons/items at the WAVE Documentation page.

## Did My Page Pass WAVE? Is it "WAVE Approved"?

Only humans can determine whether a web page is accessible. While WAVE can identify errors - if you see a red icon, your page almost certainly has an accessibility issue - it cannot tell you if your page is accessible. For this reason, we never indicate that your page is accessible or if it has 'passed' WAVE. While we invite you to provide a link on your page to the WAVE report (or simply link to http://wave.webaim.org/refer), we do not provide any approval or certification indicators or badges.

## What Guidelines and Standards Does WAVE Use? Section 508, WCAG, or Both?

A mapping of WAVE items to WCAG 2.0 success criteria is available.

WAVE supports optimal accessibility. We have added numerous tests for accessibility, including many checks for compliance issues found in the Section 508 and WCAG 2.0 guidelines. But WAVE cannot check all of the issues in these guidelines - no automated tool can. WAVE also checks some issues that extend beyond these guidelines that we know impact end users with disabilities. The documentation for each icon will indicate any relevant WCAG guidelines.

WAVE helps its users better evaluate the things that automatic tools cannot check. For example, WAVE cannot tell you if your alternative text is equivalent and appropriate, so it instead reveals the alternative text so it can be evaluated by the WAVE user. Using WAVE can help you determine if your site is both accessible AND compliant.

## Styles and Code

WAVE provides functionality for disabling page CSS styles. Complex, CSS-powered layouts may become difficult to read, especially after the WAVE icons have been embedded, so disabling styles can simplify the page presentation. Disabling styles also allows you to view the underlying reading and navigation order, the order in which keyboard-only and screen reader users will access the page.

To view the relevant code for a particular WAVE icon, you can select the Code tab at the bottom of the page to view the underlying markup/DOM of your page.

## How is the WAVE Extension Different?

The WAVE Chrome and Firefox Extensions evaluate content as it is rendered within your web browser. This means that you can evaluate private, intranet, password protected, dynamically generated, or scripted web content. All evaluation happens directly within your browser. The extension also evaluates content after scripting has been applied, whereas the server version of WAVE may not be able to apply all scripting in the page.

## The WAVE Report is 'Broken' or Difficult to Read

Because WAVE injects icons into your web page, it often breaks the page layout or results in page elements overlapping. This can typically be resolved by disabling Styles to linearize the WAVE report.

## WAVE Reports an Error, but I Don't See a Red Icon

WAVE evaluates all content within your page, not just visible content. It's most likely that the invisible WAVE error icon is within an area of your page that is visibly hidden using CSS. Such icons are typically indicated with a distinct style in the sidebar. Disabling Styles will allow you to view the icon.

## Javascript and CSS

WAVE evaluates your content after JavaScript and CSS has been applied to it. This provides a more accurate representation of actual page accessibility. However, due to security limitations, all of a page's Javascript code may not be fully applied when the page is evaluated on the WAVE web site. To best evaluate complex, scripted content, please use the WAVE Chrome or Firefox Extension.

## Is WAVE Accessible?

We have designed WAVE to be very accessible to users with disabilities, including screen reader users. Additional documentation on the accessibility features of WAVE and tips for navigating WAVE using a keyboard and assistive technology are available on the WAVE Accessibility page.

## Add a WAVE Link to Your Web Page

If you'd like to provide a link to the WAVE report for your page, simply link to http://wave.webaim.org/refer from your web page or link to http://wave.webaim.org/report?url=YOUR URL HERE to process any URL explicitly.

## Use the Browser Bookmarklet

For a quick and easy method of evaluating web pages in WAVE, you can install the following bookmarklet. It will add a WAVE item to your Bookmarks/Links/Favorites menu and/or toolbar. After installing the bookmarklet, simply browse to the web page you want to evaluate then select the bookmarklet item from the menu or toolbar. The page will be submitted to WAVE for evaluation.

WAVE this page! To install the bookmarklet, simply drag the image to the left to your Links or Bookmarks toolbar. Or you can right click on the image and select "Bookmark this link" or "Add to Favorites".

**Website and SharePoint Reporting Metrics**

The Air Force is required to submit bi-annual reporting metrics for every internet website and intranet site owned by the Air Force (Report submissions due every August and February). SAF/CNZA requires all internet and intranet site owners to provide reporting metrics to the Air Force Section 508 Coordinator one month prior to the report submission due date.

The Website Testing Worksheet was created to help streamline reporting, gather metrics, and improve the quality of data collected.

The Air Force requires a minimum of ten percent of the webpages within a website to be tested.

\*PLAN AHEAD\* Testing will require the use of ANDI and other manual techniques which can be time consuming.

The Trusted Tester Course and Certification is highly recommended for content managers.

The [Website Testing Worksheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/_layouts/15/WopiFrame.aspx?sourcedoc=/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/Website%20testing%20worksheet%20sample.docx&action=default) has been created for web content owners to evaluate the individual websites. The results are then manually transferred to the Website Testing Report which is submitted to DOD.

***It is recommended content owners use the ANDI tool to assist with their evaluation.***

**Minimum evaluation criteria**

* + Minimum of ten percent of a websites webpages must be entered
		- If a website has 10 or fewer webpages all pages must be evaluated
	+ 70 percent conformance is required to pass

## How to Fill Out the [Website Testing Worksheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/_layouts/15/WopiFrame.aspx?sourcedoc=/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/Website%20testing%20worksheet%20sample.docx&action=default):

### **Part 1: Enter Testing Info**



The DOD Bi-Annual Report requires the Air Force to report on the total number of webpages.

This section of the worksheet records the number and percentage of webpages tested on a website with a minimum of ten percent.

### **Part 2: Enter Conformance Results**

Column 1: Enter the total number of pages tested which met conformance

Each section of content is broken down to calculate a final percentage which is recorded on the Website Conformance Worksheet.

Column titled “Number of Pages which meet conformance” – enter the total number of pages which successfully passed the testing for each criteria.

#### Column 2: Enter the percentage of tested pages which met conformance.



#### Column titled “Percentage of Conformant Pages” – Manually calculate the percentage of conformant page (Number of Webpages Tested/ Number of Pages which meet conformance).

#### At the end of each table: Enter the mean of the averages listed in column 2



## Row Titled Total Average Percentage of Conformance – Enter the total Average of percentages found in the Column Percentage of Conformant Pages ((70 + 60 + 70 + 70)/4).

## This total average is placed on the Website Conformance Worksheet.

## Website and SharePoint Conformance Reporting:

The evaluations obtained from the [Website and SharePoint Testing Worksheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/_layouts/15/WopiFrame.aspx?sourcedoc=/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/Website%20testing%20worksheet%20sample.docx&action=default) must be transferred to the [Conformance Spreadsheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/website%20testing%20reporting%20sample.xlsx?Web=1). These are the final results required bi-annually.

### **How to Enter Results:**

#### Step 1: Enter website URL in Column A on [Website Conformance Reporting Spreadsheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/website%20testing%20reporting%20sample.xlsx?Web=1):



#### Step 2: Locate appropriate testing category from the [Website Testing Worksheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/_layouts/15/WopiFrame.aspx?sourcedoc=/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/Website%20testing%20worksheet%20sample.docx&action=default) (Auto-Playing and Auto-updating content) on the [Website Conformance Reporting Spreadsheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/website%20testing%20reporting%20sample.xlsx?Web=1) and enter the information in the Total Average Percentage of Conformance.





#### Step 3: Follow step 2 to enter information in Columns C-R on the [Website Conformance Testing Spreadsheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/website%20testing%20reporting%20sample.xlsx?Web=1)

#### Step 4: Column S will automatically calculate the Conformance percentage.

* An average of 70 percent or higher is required to pass

#### Step 5: Air Force Section 508 Coordinator will collect all data and total the number of conformant websites according to the [Testing Conformance Spreadsheet](https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/website%20testing%20reporting%20sample.xlsx?Web=1) and will submit results for the bi-annual report.

# **Creating Accessible PDFs**

# Workflow for creating accessible PDFs

At a high level, the process of creating accessible PDFs consists of a few basic stages:

1. Consider accessibility before you convert a document to PDF.
2. As needed, add fillable form fields and descriptions, and set the tab order.
3. Add other accessibility features to the PDF.
4. Tag the PDF.
5. Evaluate the PDF and repair tagging problems.

These stages are presented in an order that suits most needs. However, you can perform tasks in a different order or iterate between some of the stages. In all cases, first examine the document, determine its intended purpose, and use that analysis to determine the workflow that you apply.

***Consider accessibility before you convert a document to PDF***

Whenever possible, think about accessibility when you create the source files in an authoring application, such as a word-processing or page-layout application.

## Typical tasks in the authoring application include:

1. Adding alternate text to graphics
2. Optimizing tables
3. Applying paragraph styles or other document-structure features that can be converted to tags
4. Adding fillable form fields and descriptions
5. Setting the tab order
	* If your PDF includes form fields, use Tools > Accessibility > Run Form Field Recognition to detect form fields and make them interactive (fillable).

Use the Forms tools to create fillable form fields, such as buttons, check boxes, pop-up menus, and text boxes. When you create a field, type a description in the Tooltip box in the Properties dialog box for that field. Screen readers read this text aloud to the user.

Note:

You can also use the Reading Order tool in Acrobat Pro to add descriptions to form fields.

## Add other accessibility features to the PDF:

1. Set the document language
2. Make sure that security settings don’t interfere with screen readers
3. Create accessible links
4. Add bookmarks

### **Tag the PDF**

Improve the accessibility of PDFs by adding tags in Acrobat. If a PDF doesn’t contain tags, Acrobat attempts to tag it automatically when users read or reflow it, and the results may be disappointing. With a tagged PDF, the logical structure tree sends the contents to a screen reader or other assistive software or hardware in an appropriate order.

*For best results, tag a document when converting it to PDF from an authoring application.*

* Tagging during conversion to PDF requires an authoring application that supports tagging in PDF.
* Tagging during conversion enables the authoring application to draw from the paragraph styles or other structural information of the source document to produce a logical structure tree.
	+ The logical structure tree reflects an accurate reading order and appropriate levels of tags.
	+ This tagging can more readily interpret the structure of complex layouts, such as embedded sidebars, closely spaced columns, irregular text alignment, and tables.
* Tagging during conversion can also properly tag the links, cross-references, bookmarks, and alternate text (when available) that are in the file.

To tag a PDF in Acrobat:

* Choose Tools > Accessibility > Add Tags To Document.
	+ This command works on any untagged PDF, such as one created with Adobe PDF Printer. Acrobat analyzes the content of the PDF to interpret the individual page elements, their hierarchical structure, and the intended reading order of each page. Then, it builds a tag tree that reflects that information. It also creates tags for any links, cross-references, and bookmarks that you added to the document in Acrobat.
* The Add Tags To Document command
	+ Adequately tags most standard layouts. However, it cannot always correctly interpret the structure and reading order of complex page elements. These elements include closely spaced columns, irregular text alignment, nonfillable form fields, and tables that don’t have borders. Tagging these pages by using the Add Tags To Document command can result in improperly combined elements or out-of-sequence tags. These issues cause reading order problems in the PDF.

### **Watermarks and Screen Readers**

* You can add a watermark to a tagged PDF without adding it to the tag tree. Not having a watermark appear in the tag tree is helpful for individuals who are using screen readers, because they won’t hear the watermark read as a document content.
* The best way to add a watermark that doesn’t interfere with screen readers is to insert an untagged PDF of the watermark into a tagged PDF.

### **Evaluate the PDF and Repair Tagging Problems**

* Once you have a tagged PDF, evaluate the document for reading order problems, tagging errors, and accessibility errors, and then repair them as needed.
* Whichever method you use to tag the PDF, use Acrobat to touch up the tagging and reading order for complex page layouts or unusual page elements.
	+ For example, the Add Tags to Document command can’t always distinguish between instructive figures and decorative page elements such as borders, lines, or background elements. It may incorrectly tag all of these elements as figures. Similarly, this command may erroneously tag graphical characters within text, such as drop caps, as figures instead of including them in the tag that represents the text block. Such errors can clutter the tag tree and complicate the reading order that assistive technology relies on.
* If you tag a document from within Acrobat, the application generates an error report after it completes the tagging process. Use this report as a guide to repair tagging problems. You can identify other tagging, reading order, and accessibility problems for any PDF by using the Full Check/Accessibility Check tool or the Reading Order tool.

## Create a Tagged PDF from a Web Page

A PDF that you create from a web page is only as accessible as the HTML source that it is based on. For example, if the web page relies on tables for its layout design, the HTML code for the table may not flow in the same logical reading order as a tagged PDF would require, even though the HTML code is sufficiently structured to display all the elements correctly in a browser.

Depending on the complexity of the web page, you can do extensive repairs in Acrobat Pro by using the Reading Order tool or editing the tag tree in Acrobat.

To produce the most accessible PDFs from web pages you create:

* First establish a logical reading order in their HTML code. For best results, employ the Web Content Accessibility Guidelines that are published by the World Wide Web Consortium (W3C).
	1. Do one of the following:
		1. In Acrobat, choose File > Create > PDF from Web Page, enter the web page address, and then click Settings.
		2. In Microsoft Internet Explorer, in the Adobe PDF toolbar, click the Down Arrow on the Convert button and choose Preferences.
		3. In the General tab, select Create PDF Tags, and then click OK.
		4. Specify any other options as appropriate, and then click Create.
		5. Creating a tagged PDF from an authoring application

\*In most cases, you create tagged PDFs from within an authoring application, such as Adobe FrameMaker®, Adobe InDesign, or Microsoft Word. Creating tags in the authoring application generally provides better results than adding tags in Acrobat. PDFMaker provides conversion settings that let you create tagged PDFs in Microsoft Excel, PowerPoint, and Word.

## About Tags in Combined PDFs

* You can combine multiple files from different applications in one operation to create a single PDF.
	+ For example, you can combine word-processing files with slide presentations, spreadsheets, and web pages. Choose File > Create > Combine Files into A Single PDF.
* During conversion, Acrobat opens each authoring application, creates a tagged PDF, and assembles these PDFs into a single tagged PDF. The conversion process doesn’t always correctly interpret the document structure for the combined PDF because the files being assembled often use different formats. Use Acrobat Pro to create an accessible PDF from multiple documents.
* When you combine multiple PDFs into one tagged PDF, it is a good idea to retag the combined document. Combining tagged and untagged PDFs results in a partially tagged PDF that isn’t accessible to individuals with disabilities. Some users, such as those using screen readers, are unaware of the pages that don’t have tags. If you start with a mix of tagged and untagged PDFs, tag the untagged files before proceeding. If the PDFs are all untagged, add tags to the combined PDF after you finish inserting, replacing, and deleting pages.
* When you insert, replace, or delete pages, Acrobat accepts existing tags into the tag tree of the consolidated PDF in the following manner:
* When you insert pages into a PDF, Acrobat adds the tags (if any) for the new pages to the end of the tag tree. This order occurs even if you insert the new pages at the beginning or the middle of the document.
* When you replace pages in a PDF, Acrobat adds the tags (if any) from the incoming pages to the end of the tag tree. This order occurs even if you replace pages at the beginning or the middle of the document. Acrobat retains the tags (if any) for the replaced pages.
* When you delete pages from a PDF, Acrobat retains the tags (if any) of the deleted pages.
* Pages whose tags are out of order in the logical structure tree can cause problems for screen readers. Screen readers read tags in sequence down the tree, and possibly do not reach the tags for an inserted page until the end of the tree. To fix this problem, use Acrobat Pro to rearrange the tag tree. Place large groups of tags in the same reading order as the pages themselves. To avoid this step, plan on inserting pages to the end of a PDF, building the document from front to back in sequence. For example, if you create a title page PDF separately from the content, add the content PDF to the title page PDF, even though the content document is larger. This approach places the tags for the content after the tags for the title page. It’s unnecessary to rearrange the tags later in Acrobat Pro.
* The tags that remain from a deleted or replaced page don’t connect to any content in the document. Essentially, they are large pieces of empty tag tree sections. These redundant tags increase the file size of the document, slow down screen readers, and can cause screen readers to give confusing results. For best results, make tagging the last step in the conversion process. Use Acrobat Pro to delete the tags of deleted pages from the tag tree.

## About Tools for Creating Accessible PDF Forms

* Open untagged or tagged PDF forms (except PDF forms that are created from Adobe Designer) to add fillable form fields, such as text boxes, check boxes, and buttons. Then use the application’s other tools to make the form accessible. Add descriptions to form fields, tag untagged forms, set the set tab order, manipulate tags, and perform the other PDF accessibility tasks.

## Authoring Applications

* Most authoring applications that you can use to design forms don’t retain their fillable form fields when you convert the files to PDF. Use the forms tools in Acrobat Pro to add fillable form fields. Moreover, if you tag the form during conversion to PDF, the authoring application can generate inappropriate tags for the text labels of the form fields. In a complex form, for example, the text labels for all the fields can run together into a single line. Screen readers can’t interpret these fields as individual labels. Such reading order problems can require time-consuming work in Acrobat Pro to split the labels apart. In this case, producing an untagged PDF form from the authoring application is sometimes the better course. You can then use the Forms tools in Acrobat Pro to add fillable form fields before you tag the entire document. Some forms are straightforward enough that you can produce a tagged PDF from the authoring application. Then perform light touch-up in Acrobat Pro after you add the fillable form fields.

## Workflow for Creating Accessible PDF Forms

* Using Acrobat, you can open untagged and tagged PDF forms, add fillable form fields, add field descriptions and alternate text, set the tab order, and tag the forms (if they aren’t already tagged). You can also edit the tags of any tagged PDF form by using the Reading Order tool or the tag tree.

## Design the Form for Accessibility

* Forms tend to have relatively complex layouts compared to documents that have a simple, single-column structure. The success that an application has in analyzing and tagging a form depends largely on the original formatting and layout of a document, and the types of fields that it uses.
* When you design a form, include headings, instructions, and fields in which users are to enter data. At a minimum, give each field a label. Also add special instructions for fields that need them. Use graphics tools to draw lines and boxes. Don’t use characters, such as underscores and vertical bars, because these text characters can confuse screen readers.
* Adding descriptions to form fields enables screen readers to identify the fields to users. Users hear the description read aloud when they tab to the field. Write descriptions that are terse but complete. For example, the description “First name” is appropriate for a first-name field. Don’t use instructions (such as “Enter first name”) as a description.

## Set and Test the Tab Order of a Form

* The tab order for form fields enables individuals with disabilities to use a keyboard to move from field to field in a logical order. In PDF forms, set the tab order to Use Document Structure. You can test the tab order of a form by using the following keyboard commands:
1. Tab to move focus to the next field
2. Shift+Tab to move focus to the previous field
3. Spacebar to select options
4. Arrow keys to select options or list items
5. Tag the PDF form and correct tagging issues.
6. If the PDF form is already tagged, use the Reading Order tool in Acrobat to tag each form field. This tool also enables you to fix any reading order problems of the text labels for the form fields. For example, you may need to split merged lines of fields into individual fields.

# **Tool Kit**

ART Tool: <https://www.section508.gov/buy/accessibility-requirements-tool>

Sample Procurement Language: <https://buyaccessible.gov/sample-procurement>

Defined Accessibility Criteria in Contracts: <https://section508.gov/buy/define-accessibility-criteria>

Web Content Accessibility Guidelines (WCAG 2.0)(Level A/AA): <http://www.w3.org/TR/WCAG20>

Website Conformance Testing Spreadsheet: <https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/website%20testing%20reporting%20sample.xlsx?Web=1>

Website Testing Worksheet: <https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Testing%20Metrics/Website%20testing%20worksheet%20sample.docx?Web=1>

AFI 33-393, The Air Force Electronic and Information Technology Accessible to Individuals with Disabilities, Section 508: <https://static.e-publishing.af.mil/production/1/saf_cn/publication/afi33-393/afi33-393.pdf>

Creating an Accessible SharePoint Site: <https://support.office.com/en-us/article/accessibility-features-in-sharepoint-online-f291404a-dc7e-44de-a31f-d81b3099c2b9#bkm_accessiblesites>

Recommended training on Section 508:

<https://www.section508.gov/508-training/courses/508-basics/>

<https://www.section508.gov/508-training/courses/procurement/index.html>

SAF/CNZA Section 508 Org Box: usaf.pentagon.saf-cn.mbx.af-section-508@mail.mil

SAF/CNZA Section 508 SharePoint: <https://cs2.eis.af.mil/sites/10440/InfoAcc/Section%20508/Forms/AllItems.aspx>

***For further questions, contact the Air Force Section 508 Coordinator:***

Mia Day

Air Force Section 508 Coordinator

SAF/CNZA

(703) 697-4593; DSN 225-4593

mia.k.day.civ@mail.mil

Terry Bell

Section 508 Contractor Support

SAF/CNZA

(703) 695-6127; DSN 225-6127

terry.a.bell18.ctr@mail.mil