

Kaffeinated Kodemonkey
Kodemonkey Support

A11y Evaluation Audit Process for Websites

Table of Contents

About This Document.....	1
Who Should Use this Document.....	1
Harmonized Baseline Alignment.....	1
How this Document is Structured.....	1
Web Content Tests Only.....	1
Testing Order.....	2
Issues Not Covered in This Test Process.....	2
The Rationale for Each Test.....	2
Test Environment.....	3
Testing Tools.....	3
Operating Systems.....	3
Browsers.....	3
Conformance Reporting requirements.....	5
Section 508 Conformance Tests.....	6
1. Conforming Alternate Version (CAV) and Non-Interference.....	7
2. Auto-Playing and Auto-Updating Content.....	13
3. Flashing.....	18
4. Keyboard Access and Focus.....	19
5. Forms.....	25
6. Links.....	32
7. Images.....	35
8. Adjustable Time Limits.....	41
9. Repetitive Content.....	43
10. Content Structure.....	47
11. Language.....	51
12. Page Titles, Frames, and iFrames.....	54
13. Sensory Characteristics and Contrast.....	57
14. Tables.....	61
15. CSS Positioning.....	65
16. Pre-Recorded Audio-Only, Video-Only, and Animations.....	66
17. Synchronized Media.....	69
18. Resize Text.....	75

19. Multiple Ways.....	76
20. Parsing.....	77
Appendix A: Test Process Mapping.....	78
Test to Section 508/WCAG Requirement and Baseline Test (cross-reference table)	78
Section 508/WCAG Requirement to Certified Trusted Tester Test and Baseline Test (cross-reference table).....	81
Appendix B: Document Change Log.....	83
Appendix C: Test Process Quick Reference.....	90
Quick Reference with Test Conditions.....	90
One-Page Quick Reference – Test Names Only.....	94
Appendix D: ANDI Workarounds.....	95

About This Document

Who Should Use this Document

This document has is intended for use by Certified Trusted Testers in the mentorship or internship program with Kaffeinated Kodemonkey.

A Certified Trusted Tester is a person who has passed the DHS Certified Trusted Tester Certification Exam and is therefore certified to provide accurate and repeatable WCAG 2.1 & Section 508 conformance test results for web content.

A Certified Trusted Tester follows our evaluation testing process for web and uses approved testing tools, and evaluates web applications for conformance with WCAG 2.1 & Section 508 requirements.

How this Document is Structured

- Introductory Content:
 - o About this document – describes the purpose, audience, and scope of this document Section 1
 - o Test Environment – describes the test environments supported by this test process including Operating Systems, browsers, and testing tools.
 - o Conformance Reporting requirements – provides guidance
- Conformance Tests – details the test processes. Each test process includes step-by-step instructions on how and what to test, as well as instructions on how to determine test results for each Test Condition.
- Appendices – Provide cross-references tables to indicate the relationship of the Tests to WCAG 2.1 & Section 508 requirements and Baseline tests; definitions; a document change log; and quick reference summaries of the test process.

Web Content Tests Only

This test process covers web content only. Certified Trusted Tester versions 4.x and older were developed for the original Section 508 standards, which had separate requirements for web and software. With the WCAG 2.1 & Section 508 requirements applying WCAG 2.1 Level A and AA to web, software and other electronic content, combining software and web in one test process was the original plan.

However, because the WCAG 2.1 & Section 508 requirements have other requirements for software in addition to WCAG 2.1, and software testing tools were not yet available, the software test process will be separate. While it is not as common due to HTML5 capabilities, software elements are still found with web content in web applications. To test the software elements, use the software test process.

Similarly, any other operating systems, browsers, or platforms such as mobile tablets, must be evaluated using other testing procedures.

Testing Order

The numbering of the tests within the test process do not necessarily indicate the order in which tests must be performed. Each tester and each application may determine the optimal testing order for coverage and productivity.

It is recommended, however, that the first test performed is Test 1 Conforming Alternate Version. Identifying conforming alternate versions of content helps define the scope of testing and avoids unnecessary testing. Non-conforming content that has a conforming alternate version is excluded from testing.

Test 2 Auto-Playing and Auto-Updating Content and Test 3 Flashing are next in the test process, followed by Test 4 Keyboard Access and Focus. These test WCAG success criteria that are covered in [Conformance Requirement 5 Non-Interference](#). Failure to meet these success criteria could interfere with any use of the page and may indicate critical accessibility issues.

The test process was designed to streamline the sequence for testers; however, testers may choose to do the tests (after Test 1) in their preferred order. Each Test Condition (after Test 1) is independent of the tests that precede, but some will reference tests that follow.

Issues Not Covered in This Test Process

Problems may be found during testing that may affect accessibility but are simply coding errors and often affect general usability for all users. An example might include a link that leads to the wrong target website. Testers may notify a developer of these issues as a comment on a report, but they do not typically result in a compliance failure as they are beyond the scope of the WCAG 2.1 & Section 508 requirements.

The Rationale for Each Test

The Certified Trusted Tester process document provided a rationale for each test based on interpretation of the WCAG 2.1 & Section 508 requirements. With the Section 508 standards refresh and adoption of the WCAG 2.1 Success Criteria, this test process relies principally on the rationale provided in WCAG 2.1. The test process also relies on accompanying Certified Trusted Tester training to provide additional description and guidance for understanding the logic that drives each test.

Each step included in this test process document includes only the information necessary to execute the test. However, the Applicable Standards section of each test references the applicable WCAG 2.1 or Section 508 standard along with a link to the applicable article from the [Understanding WCAG 2.1](#) document.

Test Environment

At the release of this document, only the operating systems and browsers specified below were validated with the test process and tools to ensure that results were consistent and accurate.

Testing Tools

The tools used in the Test Process (and Baseline tests) have been chosen based on several factors including ease of use, ease of teaching, and accuracy of results. They are also free to install and use. The tools assist the tester with verification of accessibility properties. This test process is essentially a code inspection for accessibility properties, but the tools reduce the need for a tester to view source code or have in-depth knowledge of programming languages.

ANDI

ANDI (Accessible Name & Description Inspector) is a free open-source bookmarklet, meaning that the tool does not require installation as a plugin and can be added to multiple browsers as a bookmark. ANDI is designed to help test web content for accessibility. Developed by the US Social Security Administration, ANDI is available at <https://www.ssa.gov/accessibility/andi/help/install.html>.

ANDI issues may be reported to the ANDI GitHub page:
<https://github.com/SSAgov/ANDI/issues>.

If ANDI is not working on a specific web page, please refer to Appendix D, ANDI: Additional Information for possible solutions.

Color Contrast Analyzer

The Color Contrast Analyzer (CCA) is a free open-source tool that displays the contrast ratio for two selected colors. Developed by Steve Faulkner and the Paciello Group, CCA is available at the following links:

- [CCA 2.5 for Windows](#)
- [CCA 2.4 for macOS](#)

Operating Systems

The following operating systems were validated:

- Windows 10 (desktop mode)
- macOS (with Safari only)

Although Windows 10 and macOS are the only operating systems listed, no foreseeable issues due to using another operating system have been identified. The operating system has little to no impact on web testing results and is more dependent on the browser.

Browsers

The following browsers were validated:

On Windows 10:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Microsoft Internet Explorer 11 (IE11)

On macOS:

- Safari

Use of newer versions of these browsers is acceptable unless otherwise specified on the DHS Section 508 Compliance Testing Tools website at <https://dhs.gov/508-tools>. As browsers are frequently updated, it may be possible that an update creates critical issues for test procedures or results. Known issues and modifications will be published on the website as quickly as possible.

Conformance Reporting requirements

There are 63 Test Conditions for evaluation in this test process. Each Test Condition must have a test result for testing to be considered complete.

The 41 web requirements covered in this test process are 38 WCAG 2.1 Level A and AA Success Criteria and 3 Section 508 requirements. Each Test Condition is mapped to a web requirement (see Appendix A). Some web requirement outcomes are determined by more than one Test Condition.

While it is important to report the results for each Test Condition, it is also important to report the results for each web requirement.

Kaffeinated Kodemonkey has developed an A11y Evaluation Reporting Tool (AERT) to collect tester results and generate a test report that includes results for each Test Condition, supporting screenshots and the results for all web requirements.

(Based off the DHS TT - Accessibility Compliance Reporting Tool)

Certified Trusted Tester results may have multiple audiences including developers, purchasers, internal IT management personnel, and IT project management.

Each audience has different uses for Certified Trusted Tester results so a sufficient set of information must be included to support all audiences to the extent possible.

Evaluation results must be provided at minimum the following outcomes for Test Conditions and web requirements are **PASS, FAIL, DOES NOT APPLY, or NOT TESTED.**

Any results of **FAIL** should also include clear information identifying the location of the failure and, when feasible, clear information illustrating the content or information that resulted in the **FAIL** result. When multiple instances of the same failure are found, they may be flagged as global or included individually within test results.

If a tester cannot complete a test, a note should be added to the test report indicating “This test could not be performed”, with a detailed explanation of the issue.

Section 508 Conformance Tests

Each of the Test Conditions included in each test section below is a statement that can be evaluated as **TRUE** or **FALSE**. The “How to Test” content included under each Test Condition provides instructions on how to evaluate whether content **PASSES** the condition:

- If a Test Condition is **TRUE** for the content in question, then the content **PASSES** that Test Condition.
- If a Test Condition is **FALSE** for the content in question, then the content **FAILS** that Test Condition.
- A Test Condition **DOES NOT APPLY (DNA)** only if content does not exist that meets the conditions described in the “Identify Content” instructions at the beginning of each test section.
- In very limited instances specifically described in this test process, a Test Condition is **NOT TESTED** if content exists that meets the Identify Content criteria for a test, but a test cannot be performed due to limitations in available test tools.

1. Conforming Alternate Version (CAV) and Non-Interference

Section 508/WCAG 2.1 allows the use of conforming alternate version to meet conformance requirements. When there are multiple versions of the content, only one version is required to be fully conforming as described in this section.

WCAG defines a conforming alternate version (CAV) as a version that

1. conforms at the designated level, and
2. provides all of the same information and functionality in the same human language, and
3. is as up to date as the non-conforming content, and
4. for which at least one of the following is true:
 1. the conforming version can be reached from the non-conforming page via an accessibility-supported mechanism, or
 2. the non-conforming version can only be reached from the conforming version, or
 3. the non-conforming version can only be reached from a conforming page that also provides a mechanism to reach the conforming version

This test process covers the four parts of the definition of conforming alternate up to 4.1. In this test process, it is not practical for the tester to find all of the paths to reach the non-conforming version; therefore, it is not possible to accurately validate that the non-conforming version can only be reached as described by 4.2 or 4.3. There may also be unknown paths to the non-conforming version that may be outside of the scope of testing. For these reasons, this test process assumes access from the non-conforming version (4.1 in the CAV definition) and does not cover 4.2 or 4.3.

This Conforming Alternate Version section deviates from the other tests in this process in the following ways:

- This section instructs the tester to perform tests in other sections. (See 1.A.)
- Conformance test results should be tracked and recorded under the respective Test IDs contained in the remainder of this document. At the end of this section, the tester will indicate if conforming alternate version(s) exists, using the results from this section combined with the test results from the remaining Test IDs.
- If found, a conforming alternate version essentially replaces the non-conforming versions of that content when determining the scope of conformance and reporting conformance results.

- Non-conforming content that has a conforming alternate version is only tested for the non-interference standards.

Reference this section whenever alternate versions of content are found.

The process for determining whether a conforming alternate version exists can be summarized as follows:

1. Find the version identified as the conforming alternate version.
2. For that version:
 - a. Check if the identified version **passes all tests in this test process** (sections 2-20)
 - b. Check for **equivalence** of the identified version
 - c. Check for a **conforming access mechanism**
3. Based on these test results:
 - a. If all tests pass, it is a conforming alternate version.
 - i. Test the non-conforming versions for non-interference
 - b. If any tests fail, it is not a conforming alternate version.

Although the identified version must pass all tests in Step 2 to be considered a Conforming Alternate Version, testers should continue testing even if a failure is found in Step 2.a so that any failures of 2.b or 2.c can be reported in the same report for remediation.

A test report should indicate that in cases when the identified version fails any of the tests required to be considered the conforming alternate version, the nonconforming version is still limited to only the non-interference tests. While this deviates from WCAG's Understanding Conformance, this process assumes that the identified version will be remediated to pass all the conforming alternate version requirements.

Accessible Alternate Version

Identify Content

Per [WCAG 2.1](#), **accessible versions should be identified**, e.g., through a label on the page, as a link, in documentation, user preferences, controls to modify text appearance, etc. It may be helpful to review product documentation for information about accessible versions or enabling accessibility.

Alternate versions may be provided for a part of the page, entire pages, or an entire site. A web page or site may have more than one version of content or features. It is also possible that a site has more than one instance of conforming alternate version. The following are some indications that content is provided in more than one way.

- Content is identified as the accessible version
- Instructions are provided that describe how to enable accessibility

- Multiple methods are provided to complete a task (e.g., a calendar widget and a text field are provided for a user to enter a date)
- A link's destination is an accessible alternate version or a version for assistive technology (e.g., screen reader version)
- User preferences or settings are provided to enable accessibility
- User controls exist to modify colors and text appearance

[Per WCAG's definition](#), the identified version “does not need to reside within the scope of conformance, or even on the same web site, as long as it is as freely available as the non-conforming version.” However, the scope of testing with this process is limited to **web-based alternatives** that are available on a desktop computer. Alternate versions do not include mobile applications that can only be accessed on a mobile device.

If there is more than one version and none are identified as the conforming alternate version/accessible version, assume there is no conforming alternate version and that these Conforming Alternate Version tests **DO NOT APPLY**. Perform tests 2 through 20 on all versions.

Check alt-version-conformant

Test Name	Test ID	Test Condition
alt-version-conformant	1.A	The identified version passes all applicable Test Conditions in this test process.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if

- there is only one version of content or
- if no versions are identified as the conforming alternate version/accessible version.

How to Test:

1. Enable accessibility settings (if necessary), select, and/or navigate to the version of content identified as the accessible alternate version.
2. Following this test process, test the identified version of the content for all applicable Test Conditions. Record the result for the appropriate Test ID.
 - a. If no failures are found, this may be a conforming alternate version.
 - b. If a failure is found, the identified version is not a conforming alternate version.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**; if the following is **FALSE**, then this Test Condition **FAILS**:

1. The identified version of content passes all applicable Test Conditions in this test process.

Note:

- Alternate versions may be provided to accommodate different technology environments or user groups. At least one version would need to pass all Test Conditions.

Equivalent Alternative

Check alt-version-equivalent

Test Name	Test ID	Test Condition
alt-version-equivalent	1.B	The identified version is up-to-date with the same information and functionality.

Applicability:

This Test Condition **DOES NOT APPLY** if Test ID 1.A was evaluated as **DOES NOT APPLY (DNA)**.

How to Test:

1. Continue from Test 1.A.
2. Review the content of the non-conforming version.
3. Verify that the identified version has the same information, functionality, and language as the non-conforming version.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**; if the following is **FALSE**, then this Test Condition **FAILS**:

1. The identified version provides all of the same information and functionality in the same human language as the non-conforming content.

Note:

- [Per WCAG](#), the identified version “does not need to be matched page for page with the original (e.g., the identified version may consist of more or fewer pages).”

Conforming Mechanism

Identify Content

Identify the mechanism used to access the identified version. Various mechanisms may be used to reach the identified version, such as:

- A link to the identified version or a version for assistive technology (e.g., screen reader version)
- User preferences or settings to enable accessibility for a page or the entire site
- User controls to modify colors and text appearance of the page or entire site
- Navigating to the identified accessible version of content on a page

Check alt-version-access

Test Name	Test ID	Test Condition
alt-version-access	1.C	The mechanism to reach the identified version is accessible.

Applicability:

This Test Condition **DOES NOT APPLY** if Test ID 1.A was evaluated as **DOES NOT APPLY (DNA)**.

How to Test:

1. Perform Tests 2 through 20 for the mechanism used to reach the identified version.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**; if the following is **FALSE**, then this Test Condition **FAILS**:

1. The mechanism used to reach the accessible equivalent version passes all applicable Test Conditions.

Notes:

- Applicable mechanism tests may include Links and Buttons, Headings, Forms, and/or other tests.
- The mechanism to access the conforming version should directly or indirectly indicate that it leads to the accessible version. For example, text preceding a link to the accessible version might directly state that the link leads to the accessible version. It may also be possible to “hide” non-conforming content from AT and/or exclude it from keyboard focus, thereby limiting access only to the accessible version for users with disabilities. Such an approach, however, may not be possible, depending on the content.
- The mechanism may be explicitly provided in the content or may be relied upon to be provided by either the platform or by user agents, including assistive technologies.

Non-Interference

Identify Content

The non-conforming version(s) of the content. Exclude the version identified as the accessible version.

Check non-interference

Test Name	Test ID	Test Condition
non-interference	1.D	Content in the non-conforming version(s) meets Conformance Requirement 5.

Applicability:

This Test Condition **DOES NOT APPLY** if Test ID 1.A was evaluated as **DOES NOT APPLY (DNA)**.

How to Test:

1. If necessary and/or applicable, disable accessibility features within site settings or preferences.
2. Perform **ONLY** the following tests on the non-conforming version(s) of the content:
 - a. Test ID 2.A (1.4.2-audio-control)
 - b. Test ID 2.B (2.2.2-blinking-moving-scrolling)
 - c. Test ID 2.C (2.2.2-auto-updating)
 - d. Test ID 3.A (2.3.1-flashing)
 - e. Test ID 4.C (2.1.2-no-keyboard-trap)
3. Enter test results for the appropriate Test IDs listed above.
4. Do not perform any further testing on the non-conforming version(s) of the content.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**. If the following is **FALSE**, then this Test Condition **FAILS**:

1. The results for each of the following tests are **PASS** or **DOES NOT APPLY** for all non-conforming version(s) of the content.
 - a. Test ID 2.A (1.4.2-audio-control)
 - b. Test ID 2.B (2.2.2-blinking-moving-scrolling)
 - c. Test ID 2.C (2.2.2-auto-updating)
 - d. Test ID 3.A (2.3.1-flashing)
 - e. Test ID 4.C (2.1.2-no-keyboard-trap)

Note:

- Enter a single result for all non-conforming versions of content. (**FAIL** if any of the versions fail; **PASS** if at least one result is Pass and the others are Pass or Does not apply; **DOES NOT APPLY** if all results are Does Not Apply.)
- 3.A (2.3.1 flashing) must have a test result of **DOES NOT APPLY** in order to meet the *1.D, Non-Interference* Test Condition. A test result of **NOT TESTED** does not meet the 1.D Test Condition. See Test ID 3.A for further details.
- After performing this test on the non-conforming version of content that has a conforming alternate version for the content, omit testing of the non-conforming content from the rest of testing.
- While a conforming alternate version of content might have been confirmed under Tests 1.A through 1.C, a content owner CANNOT make a claim of conformance to the Section 508 standards if any version of content fails Test ID 1.D non-interference, including content that is not otherwise relied upon to meet conformance.

Applicable Standards**Section 508/WCAG Success Criteria****Baseline requirements**

Conforming alternate version is not a requirement. Conformance requirement #1 allows non-conforming pages to be included within the scope of conformance as long as they have a "conforming alternate version". This ensures that all of the information and all of the functionality that is on the pages inside of the scope of conformance is available on conforming Web pages.

20. Conforming Alternate Versions

WCAG Conformance Requirement 3. Non-Interference: The following success criteria apply to all content on the page, including content that is not otherwise relied upon to meet conformance, because failure to meet them could interfere with any use of the page:

3. Non-Interference

- 1.4.2 - Audio Control,
- 2.1.2 - No Keyboard Trap,
- 2.3.1 - Three Flashes or Below Threshold, and
- 2.2.2 - Pause, Stop, Hide.

2. Auto-Playing and Auto-Updating Content

Auto-Playing Audio

Identify Content

Identify audio content that automatically plays (without user activation) for more than 3 seconds.

- Content of this type includes alerts, sounds, and music.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 2.A.

Check 1.4.2-audio-control

Test Name	Test ID	Test Condition
1.4.2-audio-control	2.A	The user can pause, stop, or control the volume of audio content that plays automatically.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no audio content that plays automatically for more than 3 seconds.

How to Test:

1. Determine if there is a mechanism within the *first three elements* encountered for the user to pause or stop the audio or to control the volume of only the auto-playing audio.
 - a. The browser should already have been configured to disable auto-play. (See the Test Tool Installation and Configuration Guide for instructions.)
2. Activate the mechanism.
3. Following this test process, test the mechanism for all applicable Test Conditions.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. There is a mechanism that can pause or stop the audio or control the volume of only the auto-playing audio, AND
2. The mechanism is within the *first three elements* encountered by the user, AND
3. The mechanism passes all applicable Test Conditions in this test process.

Notes:

- The Certified Trusted Tester process requires the mechanism within three elements for clear measurability. This requirement is not specified in WCAG. To meet the non-interference requirement, the mechanism can be a focusable element or text instructions at the top of the page prior to repetitive content.
- A browser's ability to disable auto-playing media and/or mute a specific tab are acceptable mechanisms to stop or control the volume of auto-playing audio content. However, not all browsers have the capability to disable auto-playing media or mute specific windows or tabs.

Moving, Blinking, and Scrolling Content*Identify Content:*

Identify visual content that:

- Starts moving, blinking, or scrolling without user activation (including videos, synchronized media, and scrolling text), AND
- Moves, blinks, or scrolls continuously for more than 5 seconds, AND
- Is not the only content on the page.

EXCLUDE content where the movement, blinking, or scrolling of the content is essential.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 2.B.

Notes:

- When displayed content moves, blinks or scrolls, but the content itself does not change, only Test 2.B, 2.2.2-blinking-moving-scrolling applies.
- When there is auto-updating content where the content itself changes (e.g., in a carousel, stock tickers, updating sports scores), Test 2.C, 2.2.2-auto-updating applies.
- If the content moves, blinks, or scrolls AND auto-updates, both test conditions (2.B and 2.C) apply.

Check 2.2.2-blinking-moving-scrolling

Test Name	Test ID	Test Condition
2.2.2-blinking-moving-scrolling	2.B	The user can pause, stop, or hide moving, blinking, or scrolling content.

Applicability:

This Test Condition DOES NOT APPLY (DNA) if there is no moving, blinking, or scrolling content that plays automatically for more than 5 seconds **OR** if the moving, blinking, or scrolling content is the **ONLY** content on the web page.

How to Test:

1. Determine if there is an evident mechanism for the user to pause, stop, or hide the content within the *first three elements* that the user would encounter OR within three elements before/after the moving/blinking/scrolling content.
2. Activate the mechanism.
3. Following this test process, test the mechanism for all applicable Test Conditions.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. There is a mechanism that can pause, stop, or hide the content, AND
2. The mechanism is either within:
 - the *first three elements* encountered by the user, OR
 - three elements before/after the moving/blinking/scrolling content,AND
3. The mechanism **PASSES** all applicable Test Conditions in this test process.

Auto-Updating Information

Identify Content

Identify content that:

- Automatically updates (the content changes without user activation) AND
- Is not the only content on the page

Content of this type includes timers, stock tickers, news carousels, and counters.

Note:

- WCAG SC 2.2.2 does not apply to auto-updating of information where the auto-updating is essential. However, the auto-updating of a stock ticker that conveys real-time information would not be considered essential (per WCAG) so it would be included in this test. It is likely that most instances of auto-update would not be essential. To avoid incorrect omission(s) of content from this test, the tester is not tasked with determining whether auto-updating is essential and should include all content that meets the Identify Content description. A Section 508 exception may be applied for essential auto-updating content; however, this is

outside the scope of the testing process. An exception for SC 2.2.2 should be considered carefully as Conformance Requirement 5: Non-Interference requires its conformance.

- When displayed content moves, blinks or scrolls, but the content itself does not change, only Test 2.B, 2.2.2-blinking-moving-scrolling applies.
- When there is auto-updating content where the content itself changes (e.g., in a carousel, stock tickers, updating sports scores), Test 2.C, 2.2.2-auto-updating applies.
- If the content moves, blinks, or scrolls AND auto-updates, both test conditions (2.B and 2.C) apply.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 2.C and 2.D.

Check 2.2.2-auto-updating

Test Name	Test ID	Test Condition
2.2.2-auto-updating	2.C	The user can pause, stop, hide, or control the frequency of automatically updating content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no auto-updating content or the auto-updating content is the only content on the page.

How to Test:

1. Determine if there is an evident mechanism for the user to pause, stop, or hide the content or to control the frequency of the update within the *first three elements* that the user would encounter OR within three elements before/after the auto-updating content.
2. Activate the mechanism.
3. Following this test process, test the mechanism for all applicable Test Conditions.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. There is a mechanism that can pause, stop, or hide the content or control the frequency of the update, AND
2. The mechanism is either within:
 - the *first three elements* encountered by the user, OR
 - three elements before/after the auto-updating content,AND
3. The mechanism passes all applicable Test Conditions in this test process.

Notification of Automatic Content Changes

Identify Content

Identify content that changes automatically on the page as part of auto-update.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 2.D.

Check 4.1.2-change-notify-auto

Test Name	Test ID	Test Condition
4.1.2-change-notify-auto	2.D	The page provides notification of each automatic update/change in content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page content does not update or change automatically.

How to Test:

1. Identify how the user is notified of the change in content.
 - a. Identify any dialogs that alert the user to changes in content.
 - i. Determine whether the dialogs provide sufficient programmatic notification of content changes.
 - b. Identify content changes that result in focus moving to the content that has changed.
 - i. Determine whether moving the focus to the content that has changed is sufficient to notify the user of the change event (e.g., by describing the change directly in the content to which the focus moved).
 - c. Identify content changes occurring in an ARIA Live Region:
 - i. Launch ANDI: structures
 - ii. Click the “live regions” link, then use the mouse to hover over any identified live region (alternatively, use ANDI’s previous/next element buttons to navigate to identified Live Regions).
 - iii. Determine whether the changing content is contained within the Live Region.

Evaluate Results:

If any of the following is **TRUE**, the content **PASSES**:

1. The page notifies the user about a change via 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.

Note:

- This is a test for notification of automatic changes to content. The testing of the content before and after the change are to be performed in other tests. Testing of changes due to user interaction are also tested elsewhere. For example, form elements that change in response to user selections are to be tested per Test ID 5.B.

Applicable Standards**Section 508/WCAG Success Criteria****Baseline requirements**

WCAG SC 2.2.2 Pause, Stop, Hide: For moving, blinking, scrolling, or auto-updating information, all of the following are true:

- Moving, blinking, scrolling: For any moving, blinking or scrolling information that (1) starts automatically, (2) lasts more than five seconds, and (3) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it unless the movement, blinking, or scrolling is part of an activity where it is essential.
- Auto-updating: For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update unless the auto-updating is part of an activity where it is essential.
- Note 2: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page [or software application], all content [in the software or] on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

WCAG SC 1.4.2 Audio Control: If any audio on a Web page plays automatically for more than 3 seconds, either a mechanism is available to pause or stop the audio, or a mechanism is available to control audio volume independently from the overall system volume level.

- **Note:** Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference. Until such time that this test process includes a test for flashing content, no definitive statement can be made regarding Conformance Requirement 5 if any flashing content is present.

WCAG SC 4.1.2 Name, Role, Value: For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be

21. Timed Events

5. Changing Content

Section 508/WCAG Success Criteria**Baseline
requirements**

programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

3. Flashing

Flashing Content

Identify Content

Visually identify any content that flashes. Flashing is content that rapidly alternates between two or more states that vary significantly in contrast.

When flashing content IS found, the result for Test ID 3.A, 2.1.3-flashing is **NOT TESTED**.

Check 2.3.1-flashing

Test Name	Test ID	Test Condition
2.3.1-flashing	3.A	If NO flashing content is found, then this Test Condition DOES NOT APPLY (DNA). If flashing content IS found, then this test should be recorded as NOT TESTED.

Note:

- Multiple requirements are specified for conforming flashing content. To determine if requirements are met, a testing tool would be very helpful but is not available at this time. The test process will be updated when a testing tool is identified. Until then, the result “Not Tested” will indicate that flashing content was found.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 3.A.

Applicable Standards

Section 508/WCAG Success Criteria	Baseline requirements
<p><u>WCAG SC 2.3.1 Three Flashes or Below Threshold</u>: Web pages do not contain anything that flashes more than three times in any one second period, or the flash is below the general <u>flash and red flash thresholds</u>.</p> <ul style="list-style-type: none"> Note: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference. Until such time that this test process includes a test for flashing content, no definitive statement can be made regarding Conformance Requirement 5 if any flashing content is present. 	<u>9. Flashing</u>

4. Keyboard Access and Focus

Keyboard Access

Identify Content

1. 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).
2. Use the mouse to identify instances where interactive elements provide information that is essential to understanding or operating the page content.

Note:

- WCAG SC 2.1.1 requirement does not apply to functions that require input that depends on the path of the user's movement and not just the endpoints. For this test process, the tester is not tasked with identifying and omitting these types of functions. The tester should include all functions that meet the Identify Content description. A Section 508 exception may be applied for certain functions; however, this is outside the scope the testing process.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 4.A to 4.H.

Check 2.1.1-keyboard-access

Test Name	Test ID	Test Condition
2.1.1-keyboard-access	4.A	All functionality can be accessed and executed using only the keyboard.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page has no user activated functionality.

How to Test:

1. Identify the functionality and essential information provided by interactive elements.
 - a. Possible ways to identify functionality include: mouse, touch screen, voice commands, and documentation of functionality (e.g., of shortcut keys).
 - b. The "title attributes" feature in ANDI: focusable elements can help identify any essential information contained in title attributes.
2. Use the keyboard to operate identified functionality and/or access the essential information: access (e.g., tab to) the element and execute (e.g., press Enter with focus on the element).
 - a. For interactive elements with title attributes, place keyboard focus on the element. If the tooltip does not appear within two seconds, keyboard focus will not reveal the title information.
 - b. If an interactive element does not have keyboard access, determine if there is another keyboard accessible method available on the page which

provides the same functionality, e.g. one of two print methods provided is keyboard accessible, etc. [See Conforming Alternate Version for further details.]

- c. If an interactive element does not provide access to essential information via keyboard interaction, determine whether the information is available elsewhere on the page (e.g., as text).

Note:

- Not all browsers visually display the title attribute as a tooltip when an element has keyboard focus.
- Shortcut keys are typically documented on the page or in the Help documentation so that they are discoverable by a user.

Evaluate Results:

If BOTH of the following are **TRUE**, the content **PASSES**:

1. All functionality can be accessed and executed using the keyboard, AND
2. All essential information can be accessed via keyboard interaction OR the information is available elsewhere on the page.

Note:

- Any changes to functionality that occur automatically or as a result of interaction with the page should be included in this test.
- Information is considered essential or required when the information is necessary to execute an action or understand information and relationships.
- Title/tooltip information that is not essential does not require keyboard access.

Check 2.1.1-no-keystroke-timing

Test Name	Test ID	Test Condition
2.1.1-no-keystroke-timing	4.B	Individual keystrokes do not require specific timings for activation of functionality.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page has no user activated functionality.

How to Test:

1. Continue from Test 4.A.
2. Determine whether there are any instances where the timing of the keystrokes is required to activate the element, e.g., the speed at which a password keystrokes are typed is part of the password authentication.
3. If there is a timing dependent functionality, determine if there is another keyboard accessible method available on the page, which does not require specific timing.

Evaluate Results:

If the following is **TRUE**, the content **PASSES**:

1. A keyboard method is provided for functionality to be activated without requiring users to perform specific timings for activation.

Check 2.1.2-no-keyboard-trap

Test Name	Test ID	Test Condition
2.1.2-no-keyboard-trap	4.C	There is no keyboard trap.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page has no components that can receive keyboard focus.

How to Test:

1. Use standard navigation keys (e.g. TAB, SHIFT + TAB, arrow keys, CTRL + TAB, etc.) to navigate through all keyboard focusable elements on the page.
2. Determine whether there are any instances where keyboard navigation becomes trapped:
 - a. Keyboard users are unable to move away from an element, e.g. using a TAB or arrow key
 - b. Keyboard access is restricted to a small section of the page with no way to navigate out of the “loop” to the rest of the page.
3. If a keyboard trap is found:
 - a. Inspect any help (contextual help, or application help) and documentation for notification of available alternate keyboard commands (e.g., non-standard keyboard controls, access keys, hotkeys) to escape/avoid the keyboard trap.
 - b. Determine whether the alternate command(s) work.

Evaluate Results:

If ALL of the following are **TRUE**, the content **PASSES**:

1. Keyboard focus can be moved away from an element using either:
 - a. Standard navigation keys
 - b. Custom keystrokes (which are documented and available to users in the application).

AND

2. Keyboard focus can be moved away from each section of the page containing elements (and are not trapped in a “loop”, preventing access to other elements on the page) by using either:
 - a. Standard navigation keys
 - b. Custom keystrokes (which are documented and available to users in the application)

Note:

- In case of a keyboard trap, continue to test interactive elements after the trap by using the mouse to bypass the trap or refreshing the page and using the keyboard to navigate backwards through the page.

Focus*Identify Content*

Use the keyboard to navigate to keyboard-accessible interface components (including drop-down menus, form fields, revealing/hiding content, tooltips, AND all interactive interface components).

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 4.A to 4.H.

Check 2.4.7-focus-visible

Test Name	Test ID	Test Condition
2.4.7-focus-visible	4.D	A visible indication of focus is provided when focus is on the interface component.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page has no elements that can receive keyboard focus.

How to Test:

1. Continue from Test 4.C.
2. Determine whether there is a visible indication of focus on the element that has keyboard focus.
 - a. When the keyboard focus is on a frame, some browsers will display a visible focus and some may not. Where a visible focus is not available on a frame, do NOT consider this a failure of the web content.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. When each interface element receives focus, there is a visible indication of focus.

Note:

- To confirm keyboard focus is on a frame when there is not visible focus: use the TAB and SHIFT + TAB combination to deduce that the keyboard focus is on the frame. When on the frame, a tab forward should move focus to the first keyboard focusable element within the frame. From there, SHIFT + TAB once to move back to the frame and another SHIFT + TAB should move focus to a keyboard focusable element before the frame. Only the frame is permitted to not have a visible focus. Be certain it is the frame that does not have a visible focus and not another element.

Check 3.2.1-on-focus

Test Name	Test ID	Test Condition
3.2.1-on-focus	4.E	When an interface component receives focus, it does not initiate an unexpected change of context.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page has no elements that can receive keyboard focus.

How to Test:

1. Continue from Test 4.D.
2. When the interface component receives focus, evaluate whether an unexpected change of context occurs, e.g., a new window is launched, or focus is moved to another interface component.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. An unexpected change of context is not initiated when an interface component receives focus.

Check 2.4.3-focus-order-meaning

Test Name	Test ID	Test Condition
2.4.3-focus-order-meaning	4.F	The focus order preserves the meaning and operability of the web page.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page has no elements that can receive keyboard focus.

How to Test:

1. Use the tab key to move focus through the page.
2. Determine if the focus order impacts the page meaning (e.g. form fields for a mailing address are presented in the expected sequence).
 - a. This is most often noticeable when focus order does not follow the logical order of operation (normally top to bottom, left to right).
 - b. It may be necessary to use the keyboard to activate trigger controls that reveal hidden content with focusable elements (e.g., menus, dialogs, modal dialog boxes, expandable tree list) to check the focus order to, from, and within the revealed content.
 - c. It may be helpful to launch ANDI: focusable elements and select tab order.
 - d. Backward focus order does not have to mirror the forward focus order. However, it must preserve the meaning and operability of the page.
3. For modal dialog boxes, keyboard focus navigating both forward and backward should remain within the modal dialog box until it is closed.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. The focus order preserves the meaning of the page, AND
2. The focus order preserves the operability of the page.

Note:

- Focus order does not necessarily need to be top to bottom, left to right.
- When the focus order does not affect meaning or operability, this test Does Not Apply.
Example: A row of icons linking to social media may not need to be navigated in a particular order.
- ANDI tab order markup may be slightly different from actual keyboard tab order in certain browsers. Always use the results from keyboard tab order.

Applicable Standards

Section 508/WCAG Success Criteria	Baseline requirements
<u>WCAG SC 2.1.1 Keyboard</u> : All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.	<u>1. Keyboard Access</u>
<u>WCAG SC 2.1.2 No Keyboard Trap</u> : If keyboard focus can be moved to a component of the [content] using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away.	
<u>WCAG SC 2.4.7 Focus Visible</u> : Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.	<u>2. Focus</u>
<u>WCAG SC 2.4.3 Focus Order</u> : If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability.	
<u>WCAG 3.2.1 - On Focus</u> : When any component receives focus, it does not initiate a change of context.	

5. Forms

Form Components

Identify Content

1. Use ANDI: focusable elements to identify any form elements on the page, e.g., buttons, text fields, radio buttons, checkboxes, read-only fields, and multi-select lists.
2. 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.

EXCLUDE disabled input elements. These do not receive keyboard focus, cannot be selected and cannot be modified.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 5.A to 5.H.

Check 3.3.2-label-provided

Test Name	Test ID	Test Condition
3.3.2-label-provided	5.A	Visual labels or instructions are provided for form elements.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page does not have form elements or all form elements are disabled.

How to Test:

1. Determine if each form element provides visual labels or instructions.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. Visual labels or instructions are provided for each form element.

Note:

- The label or instruction must be visible when the form field has focus.
- This test only determines whether visual labels or instructions are provided, regardless of accuracy. The form label is tested for a sufficient description in 5.B (2.4.6-label-descriptive).
- The programmatic association of the form instructions (text label) to the form field is tested in 5.C for 1.3.1-programmatic-label.

Check 2.4.6 -label-descriptive

Test Name	Test ID	Test Condition
2.4.6-label-descriptive	5.B	Each visual form label is sufficiently descriptive.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page does not have form elements, if all form elements are disabled, or if no visual labels are provided for form elements.

How to Test:

1. Review the visual labels and/or instructions provided for each form component/control.
2. Determine whether labels and/or instructions for form components sufficiently describe the purpose and applicable data requirements (date formats, required fields, data type, etc.).

Evaluate Results:

If both of the following are **TRUE**, then the content **PASSES**:

1. Each visual form label is sufficiently clear and descriptive, so users know what input data is expected, AND
2. Each visual button label is sufficiently clear and descriptive, so users know its function.

Note:

- An error message is not sufficient to communicate the expected format to pass this test.
- Any changes to form labels that occur automatically or as a result of interaction with the page should be included in this test.

Check 1.3.1-programmatic-label

Test Name	Test ID	Test Condition
1.1.1-programmatic-label	5.C	The combination of the accessible name, accessible description, and other programmatic associations (e.g., table column and/or row associations) describes each input field and includes all relevant instructions and cues (textual and graphical).

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page does not have form elements.

How to Test:

1. Launch ANDI: focusable elements (this is the default selection).
2. Use the mouse or ANDI's next/previous element buttons to highlight each focusable form element and review the ANDI output.
3. Review the ANDI Output for each focusable form field.
4. If the ANDI Output does not adequately define the form element, review other programmatic associations, such as table headings or location in a hierarchical list structure, to determine whether they provide or contribute to the form component's description, cues, or instructions.
 - a. In cases where the purpose of the button is intentionally vague or ambiguous (e.g., the content to be revealed after selecting a link to "Door 1," "Door 2," or "Door 3" is intended to be a surprise), it may be sufficient for the combination of button text, accessible name, accessible description, and/or button context to refer to the button purpose vaguely or ambiguously.

Evaluate Results:

If any of the following is **TRUE**, then the content **PASSES**:

1. The ANDI Output includes all relevant instructions and cues for the form element, including when fields are required, OR
2. Descriptive labels and cues are provided by other programmatic associations (e.g., table column and/or row associations), OR
3. A combination of ANDI Output AND other programmatic association includes all relevant instructions and cues, OR
4. The combination of the programmatically determined button context and the ANDI Output provide adequate description of each buttons' purpose.

Note:

- This test also covers the requirement for [WCAG SC 4.1.2 Name, Role, Value](#).
- Any changes to form elements that occur automatically or as a result of interaction with the page should be included in this test.
- To evaluate labels and cues provided by other programmatic associations, it may be necessary to perform other tests, including but not limited to 14. Tables and 10. Content Structure.
- At minimum, radio buttons and checkboxes should be programmatically associated with their question and response.
- Form fields are not required to have programmatic associations with form section headings unless there is significant risk of confusion.

Check 3.2.2-on-input

Test Name	Test ID	Test Condition
3.2.2-on-input	5.D	Changing field values/selections (e.g., entering data in a text field, changing a radio button selection) does NOT initiate an unexpected change of context.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page does not have form elements.

How to Test:

1. Use the keyboard to navigate to form elements, e.g. text fields, radio buttons, checkboxes, buttons.
2. Complete the form element, e.g., select the radio button or check box, type information into the text box, select an item from the drop down.
3. Exit (tab away from) the completed form element and determine whether there are any instances of an unexpected change of context.
4. Changes in context include changes of: user agent, viewport, focus, content that changes the meaning of the page, e.g., a form is automatically submitted when exiting a field, a new window is launched when a radio button is selected.
 - a. **Note:** A change is not considered unexpected if:
 - i. The user is notified that a change of context is about to occur.
 - ii. The control is clearly intended to initiate a change in context when activated.

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. Changing the value of a form element does not initiate an unexpected context change.

Note:

- For some types of form fields, such as text input fields, it may be necessary to move focus away from the field to trigger an input event.

Check 4.1.2-change-notify-form

Test Name	Test ID	Test Condition
4.1.2-change-notify-form	5.E	The page provides notification of each form-related change in content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page does not have form elements or if the page content does not change due to form interaction.

How to Test:

1. Continue from Test 5.D.
2. If necessary, repeat the interactions that trigger changes to content of the page (instructions changed, error notification, content removed, content is added, button name, etc.).
3. Identify how the user is notified of the change in content.
 - a. Determine whether the form element that triggers the change has an accessible name, accessible description and/or context that provides sufficient description of the interface component's purpose.
 - i. If content changes are the direct result of a user's action while interacting with content AND the interface component that triggers the change provides sufficient description of the change, then no additional programmatic event notification is necessary.
 - b. Identify any dialogs that alert the user to changes in content.
 - i. Determine whether the dialogs provide sufficient programmatic notification of content changes.
 - c. Identify content changes that result in focus moving to the content that has changed.
 - i. Determine whether moving the focus to the content that has changed is sufficient to notify the user of the change event (e.g., by describing the change directly in the content to which the focus moved).
 - d. Identify content changes occurring in an ARIA Live Region:
 - i. Launch ANDI: structures
 - ii. Click the "live regions" link, then use the mouse to hover over any identified live region (alternatively, use ANDI's previous/next element buttons to navigate to identified Live Regions).

- iii. Determine whether the changing content is contained within the Live Region.

Evaluate Results:

If any of the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. The user's action directly results in the change in content, AND the interface component that triggered the change provided sufficient description about the change event, OR
2. The page notifies the user about a change via a keyboard-accessible dialog, OR
3. The page moves focus to the content that has changed, AND the content that has changed provides sufficient description about the change, OR
4. The content that has changed is contained in an ARIA Live Region.

Note:

- All form elements that changed during this test are to be tested per Test ID 5.C.
- It may be necessary to use the mouse to determine whether state changes occur on hover or on click.
- Depending on the component, a change of state may be triggered by various actions, such as changing values or states of other components, toggling a function, entering data in the component, mouseover, etc.
- This list of techniques to meet SC 4.1.2 is not all-inclusive; there may be emerging techniques not explicitly spelled out that are acceptable.

Input Error Identification and Suggestions*Identify Content*

Identify all automatic input error detection, error notifications, error suggestions, and related instructions:

1. Use ANDI to identify any form elements on the page.
2. 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.
3. Intentionally enter values and/or make selections that violate format and/or other form instructions to reveal automatic notifications of input errors.

If there is no automatic input error detection, the result for the following test ID(s) is **DOES NOT APPLY**: 5.F and 5.G.

Check 3.3.1-error-identification

Test Name	Test ID	Test Condition
3.3.1-error-identification	5.F	The item in error is identified in text and sufficiently described to the user in text.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the form element does not have automatic error detection.

How to Test:

1. Intentionally violate formatting and other form instructions, e.g., leave a required form field empty, use a different date format than is required, and/or create a password that does not meet the password strength requirements.
2. Attempt to submit the form and/or move to the next page.
3. Determine whether the error is identified and described in text.
 - a. The form field with the error is identified in text, e.g. "Error: Password field."
 - b. Text describes the error, e.g., in a dialog message that states "the Password you entered is incorrect."

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. The item that is in error is identified in text and sufficiently described to the user in text.

Note:

- The error message may be tested as part of 4.1.2-change-notify-forms (Test ID 5.E).

Check 3.3.3-error-suggestion

Test Name	Test ID	Test Condition
3.3.3-error-suggestion	5.G	Guidance (e.g., suggestion for corrected input) is provided about how to correct errors for form fields.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if any of the following conditions apply to the form element:

- There is no automatic input 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.

How to Test:

1. Continue from Test 5.F.
2. Determine whether guidance provides sufficient details for how to correct the error and/or offers suggestions of corrected input.

Evaluate Results:

If ANY of the following are **TRUE**, then the Test Condition is **TRUE** and the content

PASSES:

1. Suggestions for corrected input are provided, OR
2. The description contains adequate information for the user to know what is required to fix the error.

Input Error Prevention*Identify Content*

Identify Content that:

- Submits user form entries that result in or causes legal commitments or financial transactions
- Submits user form entries that modify or deletes user-controllable data in a data storage system
- Submits user test responses

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 5.H.

Check 3.3.4-error-prevention

Test Name	Test ID	Test Condition
3.3.4-error-prevention	5.H	The web page allows the user to check, reverse, and/or confirm submission.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page does not do any of the following upon submission:

- Cause legal or financial obligations.
- Modify or delete user-controlled data in data storage systems.
- Submit test responses.

How to Test:

1. Complete the required form fields with intentional errors and submit the content.

Evaluate Results:

If any of the following is **TRUE**, the content **PASSES**:

1. The user can reverse the submission, OR
2. The user is presented with an option to review, confirm, and correct information before finalizing the submission, OR

3. The page checks data for input errors and allows the user an opportunity to correct any errors.

Applicable Standards

Section 508/WCAG Success Criteria

Baseline requirements

WCAG SC: 1.1.1. Non-Text: All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for [specific] situations listed.

10. Forms

WCAG SC 2.4.6 Headings and Labels: Headings and labels describe topic or purpose.

WCAG SC 3.2.2 On Input: Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component.

WCAG SC 3.3.1 Error Identification: If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text.

WCAG SC 3.3.2 Labels or Instructions: Labels or instructions are provided when content requires user input.

WCAG SC 3.3.3 Error Suggestion: If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.

WCAG SC 3.3.4 Error Prevention (Legal, Financial, Data): For Web pages [or software] that cause legal commitments or financial transactions for the user to occur, that modify or delete user-controllable data in data storage systems, or that submit user test responses, at least one of the following is true:

1. Reversible: Submissions are reversible.
2. Checked: Data entered by the user is checked for input errors and the user is provided an opportunity to correct them.
3. Confirmed: A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.

WCAG SC 4.1.2 Name, Role, Value: For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

5. Changing Content

6. Links

Link Purpose

Identify Content

Use ANDI: links/buttons to identify all links.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 6.A and 6.B.

Check 2.4.4-link-purpose

Test Name	Test ID	Test Condition
2.4.4-link-purpose	6.A	The purpose of each link can be determined from any combination of the link text, accessible name, accessible description, and/or programmatically determined link context.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page does not have links.

Note:

- This test does not apply to links that function as an anchor or target and are not perceivable or selectable by users.

How to Test:

1. Evaluate the ANDI Output for link purpose.
2. Determine whether the ANDI Output, in combination with the programmatically determined link context (text that is in the same sentence, paragraph, list item, or table cell as the link or in a table header cell that is associated with the table cell that contains the link) adequately describes the link's purpose or function.
 - a. In cases where the purpose of the link is intentionally vague or ambiguous (e.g., the content to be revealed after selecting a link to "Door 1," "Door 2," or "Door 3" is intended to be a surprise), it may be sufficient for the combination of link text, accessible name, accessible description, and/or link context to refer to the link purpose vaguely or ambiguously.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The combination of the programmatically determined link context and the ANDI Output provide adequate description of the link's purpose.

Note:

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

Check 4.1.2-change-notify-links

Test Name	Test ID	Test Condition
4.1.2-change-notify-links	6.B	The page provides notification of each change in content that is the result of interaction with a link.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page does not have links or if the page content does not change due to link interaction.

How to Test:

1. Activate the link to trigger changes to page content (e.g., content removed, content is added, etc.).
2. Identify how the user is notified of the change in content.
 - a. Determine whether the link or button that triggers the change has an accessible name, accessible description and/or context that provides sufficient description of the interface component's purpose.
 - i. If content changes are the direct result of a user's action while interacting with content AND the interface component that triggers the change provides sufficient description of the change, then no additional programmatic event notification is necessary.
 - b. Identify any dialogs that alert the user to changes in content.
 - i. Determine whether the dialogs provide sufficient programmatic notification of content changes.
 - c. Identify content changes that result in focus moving to the content that has changed.
 - i. Determine whether moving the focus to the content that has changed is sufficient to notify the user of the change event (e.g., by describing the change directly in the content to which the focus moved).
 - d. Identify content changes occurring in an ARIA live region:
 - i. Launch ANDI: structures
 - ii. Click the "live regions" link, then use the mouse to hover over any identified live region (alternatively, use ANDI's previous/next element buttons to navigate to identified live regions).
 - iii. Determine whether the changing content is contained within the live region.

Evaluate Results:

If any of the following is **TRUE**, then the Test Condition is **TRUE** and the content

PASSES:

1. The user's action directly results in the change in content, AND the interface component that triggered the change provided sufficient description about the change event, OR
2. The page notifies the user about a change via a keyboard-accessible dialog, OR
3. The page moves focus to the content that has changed, AND the content that has changed provides sufficient description about the change, OR
4. The content that has changed is contained in an ARIA live region.

Applicable Standards**Section 508/WCAG Success Criteria****Baseline requirements**

WCAG SC 2.4.4 Link Purpose (In Context): The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.

14. Links

WCAG SC 4.1.2 Name, Role, Value: For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

5. Changing Content

7. Images

Images

Identify Content

Use the ANDI: graphics/images module to find the images. Start with the first image outlined by ANDI: graphics/images.

Use the “Focus on next element” button to find all images on the page. For each image found by ANDI: graphics/images, determine if the image is meaningful or decorative.

- All images which convey information should be considered meaningful images.
- Images that are pure decoration, used only for visual formatting, or are not presented to users, should be considered decorative. Examples of this are:
 - a swirl in the corner that conveys no information but just fills up a blank space to create an aesthetic effect
 - images used as bullet points
 - an abstract graphic used to separate sections of a page
 - an invisible image that is used to track usage statistics
 - part of a link to improve appearance or to increase the clickable area
- Images are not pure decoration if they serve to illustrate text content, suggest mood or tone, or create specific sensory experiences, such as a section dividers or bullet icons that use imagery or designs that convey specific moods or tones. If they convey any information, AT users should receive the same benefit of that information.

Note:

- ANDI may identify images that are included as a sub-component of a focusable element. In such cases, test the image using the ANDI: focusable elements module instead.
- WCAG defines “pure decoration” as “serving only an aesthetic purpose, providing no information, and having no functionality.”
- WCAG defines “specific sensory experiences” as “a sensory experience that is not purely decorative and does not primarily convey important information or perform a function.”

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 7.A to 7.B.

For Meaningful Images – Check 1.1.1-meaningful-image-name

Test Name	Test ID	Test Condition
1.1.1-meaningful-image-name	7.A	The accessible name and accessible description for a meaningful image provides an equivalent description of the image.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no meaningful images on the page.

How to Test:

1. Review the ANDI Output for the meaningful image.
 - a. The ANDI Output must provide an equivalent description of the image. This can be provided by a brief description of the image in the ANDI Output and instructions about where to obtain the full description.
 - b. If the image is used as a CAPTCHA, ANDI Output describes the purpose of the CAPTCHA.
 - c. If the image is of meaningful text, ANDI Output contains the same text.
 - d. If an image is a component or child of a focusable element (e.g., a button or a link), evaluate the accessible name for the entire focusable element, not just the image. Use ANDI: focusable elements (and/or ANDI: links/buttons if applicable) to determine the accessible name for the combined component.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The ANDI Output contains the equivalent description for the meaningful image and/or refers to a description in the page content.

Note:

- Any changes to meaningful images that occur automatically or as a result of interaction with the page should be included in this test.
 - Notification of automatic changes are tested in Test 2.D.
 - Notification of changes as a result of interaction with other content are tested either in Test 5.E or 6.B.
- An image that is on the page but not detected by ANDI as described in Identify Content should not be included in this test.
- The ANDI Output is empty for an image that has role="presentation" or aria-hidden="true". These should not be set for meaningful images, which must have an equivalent description in ANDI Output.

For Decorative Images – Check 1.1.1-decorative-image

Test Name	Test ID	Test Condition
1.1.1-decorative-image	7.B	There is no accessible name and accessible description for a decorative image.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no decorative images on the page.

How to Test:

1. Review the ANDI Output for the decorative image.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The ANDI Output for a decorative image is blank.

Note:

- An image that is on the page but not detected by ANDI as described in Identify Content should not be included in this test. Some decorative images are background images and should be tested and reported in 7.C.
- Any changes to decorative images that occur automatically or as a result of interaction with the page should be included in this test.
- The ANDI Output is empty for an image that has role="presentation" or aria-hidden="true". These are two techniques to hide an image from assistive technology and are appropriate for decorative content.

CSS Background Images

Identify content

1. Use the ANDI: graphics/images module to find CSS background images. If the “find background” and “hide background” buttons are available, background images are on the page.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 7.C.

Check 1.1.1- decorative-background-image

Test Name	Test ID	Test Condition
1.1.1- decorative-background-image	7.C	The background image is not the only means used to convey important information.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no background images on the page.

How to Test:

1. Select the “find background” button in ANDI: graphics/images to outline all background images (in green).
2. Find the outlined background images within the page. (ANDI will not display information for background images.)
3. Determine whether important information provided by the background image is available without the background image.
 - a. Select the “hide background” function in ANDI: graphics/images to hide background images and help determine if the image’s information is also available on the page without the background image.
 - b. Review the sequence or positioning of the image to determine whether equivalent information is presented in the same logical order.

Evaluate Results:

If any of the following is **TRUE**, then the content **PASSES**:

1. The background image is decorative, OR
2. The meaning of the background image is also available without the background image.

Note:

- Any changes to meaningful background images that occur automatically or as a result of interaction with the page should be included in this test.

CAPTCHA Images

Identify content

Identify all CAPTCHA images.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 7.D.

Check 1.1.1-captcha-alternative

Test Name	Test ID	Test Condition
1.1.1-captcha-alternative	7.D	Alternative forms of CAPTCHA are provided.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no CAPTCHA images on the page.

How to Test:

1. Determine whether alternative forms of CAPTCHA with output modes for different types of sensory perception are provided.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. The CAPTCHA has a format for users without vision, AND
2. The CAPTCHA has a format for users without hearing.

Images of Text

Identify content

Identify all images of text

EXCLUDE 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.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 7.E.

Check 1.4.5-image-of-text

Test Name	Test ID	Test Condition
1.4.5-image-of-text	7.E	The image of text cannot be replaced by text or is customizable.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no images of text on the page.

How to Test:

1. Determine if text can be used instead of the image of text to present the same effect and information.
 - a. Logotypes (text that is part of a logo or brand name) cannot be replaced by text.
 - b. Type samples, branding, images of specific fonts that are not widely supported are additional examples of images of text that cannot be replaced by text.
2. Determine if the image of text can be visually customized: adjust the font, size, color and background with controls provided by the web page.
 - a. Customizing font size for an image of text also implies the ability to adjust the size without pixelation (which is typically evident when simply using the browser resize functionality to resize images).

Evaluate Results:

If ANY of the following is **TRUE**, then the content **PASSES**:

1. The image of text cannot be replaced with text, OR
2. The image of text can be visually customized.

Applicable Standards

Section 508/WCAG Success Criteria

Baseline requirements

WCAG SC: 1.1.1. Non-Text: All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for [specific] situations listed.

6. Images

WCAG SC: 1.4.5 Images of Text: If the technologies being used can achieve the visual presentation, text is used to convey information rather than images of text except for [specific situation listed].

WCAG SC: 1.1.1. Non-Text: All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for [specific] situations listed.

18. CSS Content and Positioning

WCAG SC 4.1.2 Name, Role, Value: For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

5. Changing Content

8. Adjustable Time Limits

Timing Adjustable

Identify Content

Identify any instances of content time limits.

Time limits could be identified by:

- Inspecting system or site documentation
- Text description somewhere on the page where the time limit occurs
- Pop-ups or other 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.

EXCLUDE:

- **Real-time Exception:** The time limit is a required part of a real-time event (for example, an auction), and no alternative to the time limit is possible; or
- **Essential Exception:** The time limit is essential and extending it would invalidate the activity; or
- **20 Hour Exception:** The time limit is longer than 20 hours.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 8.A

Check 2.2.1-timing-adjustable

Test Name	Test ID	Test Condition
2.2.1-timing-adjustable	8.A	The user can turn off, adjust, or extend the time limit.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no time limit for content or if the time limit meets one of the exceptions listed in the Identify Content section above.

How to Test:

1. Determine whether the web page provides a way to turn off, adjust, or extend the time limit.

Evaluate Results:

If any of the following is **TRUE**, then the content **PASSES**.

1. The user can turn off the time limit before time expires, OR
2. The user can adjust the time limit to at least ten times the length of the default setting before time expires, OR
3. The page provides a warning before time expires AND:
 - a. For a period of at least 20 seconds, the user can extend the time limit with a simple action (e.g., pressing the spacebar), AND
 - b. The user can extend the time limit at least ten times.

Applicable Standards

Section 508/WCAG Success Criteria

Baseline requirements

WCAG SC 2.2.1 Timing Adjustable: For each time limit that is set by the content, at least one of the following is true:

- **Turn off**: The user is allowed to turn off the time limit before encountering it.
- **Adjust**: The user is allowed to adjust the time limit before encountering it over a wide range that is at least ten times the length of the default setting.
- **Extend**: The user is warned before time expires and given at least 20 seconds to extend the time limit with a simple action (for example, “press the space bar”), and the user is allowed to extend the time limit at least ten times.

21. Timed Events

9. Repetitive Content

Repetitive Content - Bypass

Identify Content

Identify block(s) of content that are repeated on other pages within the site.

- Blocks of content that are repeated on other pages may include navigation links, page headers, tabs, and banners.
- Blocks of content do not have to be exactly the same to be considered repetitive; blocks of content could be considered to repeat if they contain the same type of information and/or serve the same purpose.

EXCLUDE small sections, such as repeated individual words, phrases, or single links. They are not considered repetitive blocks of content.

Note:

- Most web browsers provide keyboard shortcuts to move the user focus to the top of the page or browser; providing a method to bypass a set of navigation links located at the bottom of a web page may be unnecessary.

If there is no such content, the result for the following test ID(s) is **DOES NOT**

APPLY: 9.A

Check 2.4.1-bypass-function

Test Name	Test ID	Test Condition
2.4.1-bypass-function	9.A	A keyboard-accessible method is provided to bypass repetitive content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** to a web page that does not contain blocks of content that are repeated on other web pages.

How to Test:

1. Starting at the top of the page, use keyboard commands to navigate forward to repetitive blocks of content. **Note:** Some bypass functions may not be visible until they receive focus.
2. Determine whether a keyboard-accessible method was provided to bypass repetitive content (e.g. skip links, hotkeys, scripted elements, etc. Frames may work as a bypass method in some browsers but not others).
 - a. Launch ANDI: focusable elements to check for skip links, hide options, collapse menu and other elements with similar bypass functionality.
 - i. **Note:** ANDI's "tab order" feature under the focusable elements module may help evaluate the order in which bypass methods occur relative to other content.
 - ii. Alternatively, launch ANDI: links/buttons and select the "view links list" button.
3. Use keyboard commands to activate the bypass function.

- a. Multiple blocks of repeated content may require multiple methods to bypass the blocks; it may not be possible to bypass all blocks of repeated content with a single method.
- b. Moving focus past blocks of repeated content may not always be visibly evident if there are no focusable elements directly after the bypassed block.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. There is a keyboard-accessible method provided to bypass repetitive content, AND
2. When activated, the method works, and the block of content is bypassed.

Note:

- If there is no interactive component to receive the shift of focus, it may not be evident via visual indication of focus that a focus shift occurred. Reducing the browser height may make a focus shift more obvious.
- The bypass function(s) must not skip over content that is not repetitive.

Repetitive Content - Navigation*Identify Content*

Identify all navigational elements that are repeated on multiple pages within the website.

- **Note:** Navigational elements are any components that provide a user the ability to locate specific information or functionality across the website. These can be static or interactive elements, and groupings of components can also meet this definition.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 9.B

Check 3.2.3-consistent- navigation

Test Name	Test ID	Test Condition
3.2.3-consistent-navigation	9.B	Each navigational element occurs in the same relative order with regard to other repeated components on each web page where it appears.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** to a web page that does not contain components that are repeated on other web pages.

How to Test:

1. Review multiple web pages of the web site to identify navigational components that are repeated on multiple pages. Do not initiate changes to the content.
2. Review the order of the navigational elements and compare it to the order on the other pages where they appear
 - **Note:** ANDI's "tab order" feature under the focusable elements module may help evaluate the focus order of interactive interface components. ANDI's "reading order" feature under the structure module may also help evaluate the content order of both focusable and non-focusable components.

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. Each repeated component occurs in the same relative order with regard to other repeated components on each web page where it appears.

Note:

- *Same relative order* is defined as same position relative to other items. Items are considered to be in the same relative order even if other items are inserted or removed from the original order. For example, expanding navigation menus may insert an additional level of detail or a secondary navigation section may be inserted into the reading order.

Repetitive Content - Identification

Identify Content

Identify components that have the same functionality within a set of web pages.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 9.C.

Note: This test does not apply to components that only appear within the page and do not appear on other pages.

Check 3.2.4-consistent-identification

Test Name	Test ID	Test Condition
3.2.4-consistent-identification	9.C	The accessible name and description is consistent for components that perform the same function.

Applicability:

This Test Condition only applies to components that have the same functionality within a set of web pages.

How to Test:

1. Launch ANDI: focusable elements.
2. Examine the ANDI Output for each identified element.

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. Components with identical functionality are identified consistently.

Note:

- Consistent text alternatives for interface elements that perform the same function are not always truly “identical.” This is acceptable if they follow a consistent format. For instance, in the use of a graphical arrow at the bottom of a web page that links to the next web page, the text alternative may be: “Go to page 4.” However, the same arrow image on the next page should then state “Go to page 5.”
- A single non-text-content-item may be used to serve different functions. In such cases, different text alternatives are necessary and should be used. Examples can be commonly found with the use of icons such as check marks, cross marks, and traffic signs. Their functions can be different depending on the context of the web page. A check mark icon button may function as “approve”, “mark completed”, or “include”, to name a few, depending on the situation. Using “check mark” as the text alternative across all web pages does not help users understand the function of the button. Different text alternatives can be used when the same non-text content serves multiple functions. ([Understanding SC 3.2.4](#))

Applicable Standards

Section 508/WCAG Success Criteria

Baseline requirements

WCAG SC 2.4.1 Bypass Blocks: A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.

4. Repetitive Content

WCAG SC 3.2.3 Consistent Navigation: Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user.

WCAG 3.2.4 Consistent Identification: Components that have the same functionality within a set of Web pages are identified consistently.

10. Content Structure

Headings

Identify Content

1. Identify all visually apparent headings, which denote sections of content.
 - a. Headings are often in a larger, bolded font separated from paragraphs by extra spacing (though not always). Note the hierarchy and structure of each heading with respect to other headings on the page.
2. Use ANDI to identify all programmatically defined headings: <h1> to <h6> or ARIA role="heading".
 - a. Launch ANDI: structures.
 - b. Select the "headings" button within ANDI: structures.
 - c. ANDI will add dotted outlines around each identified heading that is visible on the page.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 10.A to 10.C.

Check 2.4.6-heading-purpose

Test Name	Test ID	Test Condition
2.4.6-heading-purpose	10.A	Each heading describes the topic or purpose of its content.

Applicability:

This Test Condition **DOES NOT APPLY** if there are no visual headings on the page.

How to Test:

1. For each visually identified heading, compare the heading text to the content beneath the heading.

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. The heading describes the topic or purpose of its content.

Check 1.3.1-heading-determinable

Test Name	Test ID	Test Condition
1.3.1-heading-determinable	10.B	Each programmatically determinable heading is a visual heading and each visual heading is programmatically determinable.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** to a web page that has no programmatic headings identified by ANDI and has no visual headings.

How to Test:

1. Select ANDI: structures and review the ANDI Output for each visually apparent heading. ANDI outlines all headings with a dotted purple line.
 - a. If ANDI does not identify a visually apparent heading, then the heading is not defined programmatically.
2. Review each heading identified by ANDI to determine if it is also a visually apparent heading.
3. Review the ANDI Output for each heading to determine if it matches the visual heading. If they do not match, then the heading is not properly defined programmatically.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. Each programmatically determinable heading is serving as a visual heading on the page, AND
2. Each visual heading is programmatically defined.

Note:

- Content that is not a visual heading should not have a role of heading (for example, heading markup should not be used for emphasis on an element that is not a heading for content after it). Conversely, content that is styled to look like and function like a heading should be programmatically defined as heading.

Check 1.3.1-heading-level

Test Name	Test ID	Test Condition
1.3.1-heading-level	10.C	Programmatic heading levels logically match the visual heading presentation within the heading structure.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** to web pages where programmatic headings are not identified by ANDI.

How to Test:

1. Launch ANDI: structures and select the “view outline” button to display the Structure Outline.
2. Mouse over or tab through each of the headings in ANDI’s Structure Outline to review the ANDI Output for each heading.
 - a. ANDI will identify heading level conflicts if found. Ex: “Heading element level <h1> conflicts with [aria-level=“2”].”
3. Compare the heading levels listed in the Structure Outline to the page content. Determine whether the heading levels logically match the visual heading presentation within the heading structure.

- a. 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.
- b. 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.
- c. Headings with an equal or higher level start a new section; headings with a lower level start new subsections that are part of the higher leveled section.
- d. A heading level 1:
 - Is not required
 - Can be used more than once on a page
 - Is not required to match the page title
- e. 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 also acceptable to have `<h3>` then `<h5>` without an `<h4>` in between.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. Every programmatically identified heading level logically matches the visual heading structure on the page, AND
2. There is no heading level conflict.

Lists*Identify Content*

Identify all visually apparent lists on the page.

Note:

- Exclude menus or other elements that are part of the page's navigational framework from this test. Developers might use list elements to create grouped navigational items, such as menus, submenus, lists of navigation links, breadcrumbs, etc. while styling them to remove bullets or numbering and orienting the lists horizontally instead of vertically. If these navigational elements are part of the page's navigational framework, separate from the main content, they should not be considered visually apparent lists for this test and should be excluded. However, any lists which are part of the main content should be included.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 10.D.

Check 1.3.1-list-type

Test Name	Test ID	Test Condition
1.3.1-list-type	10.D	All visually apparent lists are programmatically identified according to their type.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no visually apparent lists.

How to Test:

1. Launch ANDI: structures and select the “lists” button.
2. Review the information under “List Elements,” noting the number of lists identified and their types.
3. For each list, visually determine the type of list; determine if it appears to be ordered, unordered, or a description list. Exclude menus and navigational framework elements from this test, unless they are part of the main content.
 - Ordered - numbered sequentially and, if necessary, hierarchically (e.g., 1, 2, 2.a, 2.a.i, etc.) and are used where sequence or the ability to reference specific items by number/letter are important.
 - Unordered - not numbered and are used where a specific sequence or the ability to reference specific items by number/letter are not important.
 - Description list (dl) - used to groups terms with their descriptions.
4. Review the visual representation of list relationships, including order, hierarchy, and nesting compared to the programmatic list definitions presented via the ANDI output.
 - a. It is possible to provide any number of nested list combinations using ordered, unordered, and definition lists. ANDI identifies each nested list separately. Review each list using the “Inspect Next Element” button to determine if the visual nesting and relationship matches the programmatic nesting and relationships.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. All content that has the visual appearance of a list is defined programmatically as a list, according to the type of list.
 - a. An unordered list (with or without bullets) is marked as an unordered list (ul).
 - b. An ordered list is marked as an ordered list (ol).
 - c. Terms and their descriptions that are presented in the form of a list are marked as a description list (dl)
- AND
2. All programmatic list relationships, including nesting and hierarchies, are consistent with the list relationships presented visually.

Note:

- Not all lists require markup. For instance, a list of items in a sentence, separated by commas, do not have to be included in a bulleted or numbered list.

Applicable Standards**Section 508/WCAG Success Criteria****Baseline
requirements**

WCAG SC 2.4.6 Headings and Labels: Headings and labels describe topic or purpose. 13. Content Structure

WCAG SC 1.3.1 Info and Relationships: Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.

11. Language

Language of Page

Identify Content

1. Identify all pages with text.
2. Review the page content to identify the default human language of the page (the language in which most of the page is presented).

Note:

- “Human language” refers to the language that people use to communicate with one-another as opposed to coding languages used to produce software and web content.

Test ID 11.A always applies.

Check 3.1.1-page-language-defined

Test Name	Test ID	Test Condition
3.1.1-page-language-defined	11.A	The default human language of each web page can be programmatically determined.

Applicability:

This Test Condition always applies – you may NOT evaluate the condition as **DOES NOT APPLY (DNA)**. All pages should have some textual content, even if that content is included programmatically as alternative text for non-text content.

How to Test:

1. Launch ANDI: structures.
2. Click the "more details" link, then “page language.”
 - a. ANDI will display a dialog listing the value of the lang attribute assigned to the <html> element of the page.
 - b. If no lang attribute is defined or if the attribute is empty, ANDI will provide a warning in the same dialog.
3. Consult the [Internet Assigned Numbers Authority's \(IANA\) Language subtag registry](#) to determine whether the language is properly defined and matches the default human language for the page.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. The default primary language is correctly specified per [IANA](#), AND
2. The identified language in the lang attribute correctly matches the default human language for the page.

Note:

- The primary language subtag is the first two- or three-character code in the value of the lang attribute. (Do not test additional language specifications for dialects that may follow the primary language subtag.)

- The primary language subtag must conform to the [Internet Assigned Numbers Authority's \(IANA\) Language subtag registry](#).

Language of Parts

Identify Content

1. Identify any text content that differs from the default human language of the page including alternative text.
2. Identify the human language of the text content that differs from the default human language of the page.

Note:

- Proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text do not require a lang attribute different from the default language of the page and are not covered by this test.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 11.B.

Check 3.1.2-part-language-defined

Test Name	Test ID	Test Condition
3.1.2-part-language-defined	11.B	The human language for any content segment that differs from the default human language of the page can be programmatically determined.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if all of the content in the page is in the same human language.

How to Test:

1. Launch ANDI: structures; click the “more details” link, then “[#] lang attributes”. If the [#] is zero, no content was marked with a lang attribute.
2. Locate the markup added to the web page that identifies the element to which the attribute is applied and the language defined in the language attribute value (e.g., “en” for English).
 - a. Mouseover or tab to the markup to reveal the beginning and end of the element.
 - b. Determine whether the entire passage is enclosed within the element.
3. Consult the [Internet Assigned Numbers Authority's \(IANA\) Language subtag registry](#) to determine whether the language is properly defined and matches the human language for the content segment.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. The language for the content segment that differs from the primary default language of the page is correctly specified per [IANA](#), AND
2. The identified language in the lang attribute correctly matches the human language for the content segment

Note:

- The primary language subtag is the first 2 or 3 character code in the value of the lang attribute. (Do not test additional language specifications for dialects that may follow the primary language subtag.)
- The primary language subtag must conform to the [Internet Assigned Numbers Authority's \(IANA\) Language subtag registry](#).

Applicable Standards**Section 508/WCAG Success Criteria****Baseline requirements**

WCAG SC 3.1.1 Language of Page: The default human language of each Web page can be programmatically determined.

15. Language

WCAG SC 3.1.2 Language of Parts: The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text.

12. Page Titles, Frames, and iFrames

Page Titles

Identify Content

All web pages.

Test Conditions 12.A and 12.B always apply.

Check 2.4.2-page-title-defined

Test Name	Test ID	Test Condition
2.4.2-page-title-defined	12.A	A <title> element is defined for the web page.

Applicability:

This Test Condition always applies – you may NOT evaluate the condition as **DOES NOT APPLY (DNA)**.

How to Test:

1. Launch ANDI: structures. Review the alerts in ANDI's "Accessibility Alerts" section to determine whether ANDI displays any of the following Invalid HTML Alerts:
 - a. "Page has no <title>"
 - b. "Page <title> cannot be empty"

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. A Page Title is defined for the web page.

Check 2.4.2-page-title-purpose

Test Name	Test ID	Test Condition
2.4.2-page-title-purpose	12.B	The <title> element identifies the contents or purpose of the web page.

Applicability:

This Test Condition always applies – you may NOT evaluate the condition as **DOES NOT APPLY (DNA)**.

How to Test:

1. Launch ANDI: structures, then select "more details", then "page title."
 - a. A modal dialog box will appear with the identified page title listed.
2. Evaluate the purpose and content of the web page.
3. Determine whether the Page Title is a meaningful representation or indication of page content.
 - a. If the web page is part of a set of web pages, determine whether the Page Title is sufficient to distinguish the web page from other pages.
 - b. For documents or web applications, the name of the document or web application would be sufficient to describe the purpose of the page.

Evaluate Results:

If ALL of the following are **TRUE**, then the Test Condition is **TRUE** and the content

PASSES:

1. The Page Title accurately identifies the contents or purpose of the web page, AND
2. If the web page is part of a set of web pages, the Page Title accurately distinguishes the web page from other pages in the web site.

Note:

- A web application is an application that runs in a web browser (such as webmail) and may not have a URL that changes as content on the web page changes.

Frames*Identify Content*

Use ANDI to identify all Frames.

1. Launch ANDI. If a Frame is used, ANDI will provide a notification that frames have been detected.
 - a. If there are no Frames, ANDI will provide no notification.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 12.C.

Check 4.1.2-frame-title

Test Name	Test ID	Test Condition
4.1.2-frame-title	12.C	Each <frame> has a title attribute that describes its content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if no Frames are identified.

How to Test:

1. Launch ANDI. If a Frame is used, ANDI will provide a notification that Frames have been detected. Select “Cancel” to test an individual Frame.
2. ANDI will list each Frame, and, if available, the associated title attribute.
 - a. If there is no title attribute, ANDI will reveal an “alert”.
3. Select each link to access each frame. Review the Frame to understand its content. Navigate back to the page being tested and launch ANDI again.
4. In ANDI, review each frame’s corresponding title attribute for a meaningful description of content.

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. All frames have a title attribute that describes its content.

Note:

- In HTML5 the <frame> element is marked as obsolete. While the <frame> element has been deprecated in HTML5, testers may still encounter web pages and/or web applications with code that, while outdated, can and should still be accessible.
- Frame content must be tested for conformance with all other applicable tests (12.A and 12.B do not apply to frame content).
 - o To open each Frame to test: Launch ANDI and select “Cancel”. A list of Frames will show; select the link to test an individual frame.

iFrames*Identify Content*

Use ANDI to identify all iframes in the tab order.

1. Launch ANDI: iframes to navigate to and highlight iframes on the page. If there is not an option for the iframes module, there are no iframes on the web page.
2. Use the “Next Element” button to find all iframes that have the following listed in the Accessibility Component:
 - a tabindex value that is not negative (e.g. 0, 1) meaning the iframe is in the tab order
 - OR
 - no tabindex shown.

Test only these <iframes>.

Note: A negative tabindex such as -1 means that the iframe is explicitly excluded from the tab order. Do not test iframes explicitly excluded from the tab order.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 12.D.

Check 4.1.2-iframe-name

Test Name	Test ID	Test Condition
4.1.2-iframe-name	12.D	The combination of accessible name and description for each <iframe> describes its content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if no iframes are identified or if the iframes are explicitly excluded from the tab order (i.e., the tabindex is a negative number).

How to Test:

1. Launch ANDI: iframes
2. Review the ANDI Output for each iframe that has a non-negative tabindex value or where the tabindex is not defined to determine whether the accessible name and description accurately describe the content of each <iframe>.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The ANDI Output for each <iframe> in the tab order sufficiently describes its content.

Note:

- All iframe content must be tested for conformance with all other applicable tests (12.A and 12.B do not apply to iframe content). To open each iframe to test: Within ANDI: iframe, select the button to “test in new tab.”
- If ANDI fails to run in an iframe, see Appendix D for possible workarounds.

Applicable Standards

Section 508/WCAG Success Criteria	Baseline requirements
WCAG SC 2.4.2 Page Titled : Web pages have titles that describe topic or purpose.	11. Page Titles
WCAG SC 4.1.2 Name, Role, Value : For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.	19. Frames and iFrames

13. Sensory Characteristics and Contrast

Use of Color

Identify Content

Identify content that relies on color to convey meaning, such as indicate an action, prompt a response, or distinguish a visual element.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 13.A.

Check 1.4.1-color-meaning

Test Name	Test ID	Test Condition
1.4.1-color-meaning	13.A	Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if no content relies on color to convey meaning.

How to Test:

1. Determine whether color is the only visual method used to convey information (e.g., review the onscreen text for a full description and/or look for other visual cues).

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. When color is used to convey information, indicate an action, prompt a response or distinguish a visual element, another visual, onscreen method is used to convey the information which does not use color.

Note:

- Alternate text that appears on mouse-over of a visual element is not considered to be “onscreen text.”
- An error indicator cannot use color alone as an indicator.
- It is considered a browser setting if a visited link changes color, and this is not failed for 13.A.

Use of Sensory Characteristics

Identify Content

Identify instructions for understanding and operating content that rely on sensory information to convey information, e.g., references to shape, color, size, visual location, orientation, or sound.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 13.B.

Check 1.3.3-sensory-info

Test Name	Test ID	Test Condition
1.3.3-sensory-info	13.B	Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components, such as shape, color, size, visual location, orientation, or sound.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page's instructions do not rely on sensory information.

How to Test:

1. Determine if the instructions using sensory characteristics provide details that allow content to be located, identified, understood, and operated without any knowledge of its shape, color, size, orientation, or relative position.
2. Check for any auditory cues that are provided as instructions for understanding and operating content.
 - a. Ensure sound is not muted while testing.
 - b. Determine whether there are visual and/or textual cues that provide the same information.

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. When instructions use shape, color, size, location, orientation, or sound to convey meaning, another method that does not rely on sensory characteristics is provided.

Note:

- The use of “above” or “below” is allowable when it references the sequential order of content.
- Part of testing 13.A for 1.4.1-color-meaning is to ensure color alone is not used to convey information. This includes if color alone is used to provide instructions or operating procedures. Neither can other sensory characteristics be used alone to provide instructions or operating procedures.

Color Contrast

Identify Content

Identify ALL text AND images of text.

EXCLUDE text that is:

- In logotypes: logo or brand name
- For inactive (disabled) user interface components
- For purely decoration purposes and not meaningful, i.e., having no functionality
- Contained within a picture that contains significant other visual content
- Changed to indicate it is a “visited” link

Note:

- Some text may not initially be visible on the page, including text that becomes visible on mouseover or when an element receives focus. Nevertheless, the text must still conform to the color contrast requirement, wherever it occurs.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 13.C.

Check 1.4.3-contrast

Test Name	Test ID	Test Condition
1.4.3-contrast	13.C	The visual presentation of text and images of text have sufficient contrast.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the page has no visible text or images of text.

How to Test:

1. Launch ANDI: color contrast.
2. Review any “Contrast Alerts” in ANDI’s “Accessibility Alerts” section to identify any text that fails to meet the minimum contrast ratio.
3. In ANDI’s “Accessibility Alerts” section, identify any “Manual Contrast Tests Needed.”
 - a. If the text is not selectable or appears on a background image, determine the contrast using the *Colour Contrast Analyser*.
 - b. Open the *Colour Contrast Analyser* tool, select the Foreground color-dropper button, and click a pixel in the text font. If the text color is varied in appearance or color, choose a pixel that provides the least contrast.
 - c. Select the Background color-dropper button, then click a pixel in the background close to the text. If the background is varied in appearance or color, choose a pixel that provides the least contrast.
 - d. Identify the Contrast Ratio

- e. Compare the contrast ratio against the minimum required contrast ratio identified in the ANDI Contrast Ratio output.
4. If the page contains an image of text alone, or an image with text and no other significant content, test the image of the text with the CCA to determine the contrast ratio between the foreground (text) and background.

Note: This is for instances where it is impossible for the ANDI: color contrast module to detect the presence of the text.

 - a. Select the graphics/images module in ANDI.
 - b. Use the ANDI arrow buttons to identify any images of text or images with text and no other significant content.
 - c. Open the CCA and test the contrast of the text in the image.
 - i. Select the Foreground color-dropper button and click a pixel in the text font. If the text color is varied in appearance or color, choose a pixel that provides the least contrast.
 - ii. Select the Background color-dropper button and click a pixel in the background close to the text. If the background is varied in appearance or color, choose a pixel that provides the least contrast.
 - iii. Determine whether the resulting contrast ratio is at least 4.5:1.

Evaluate Results:

If any of the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. The contrast between the text and its background is equal to or greater than the minimum required contrast ratio identified in the ANDI Contrast Ratio output, OR
2. If the text is an image of text, the contrast between the image of text and its background is equal to or greater than 4.5:1 as identified using the Colour Contrast Analyser.

Applicable Standards

Section 508/WCAG Success Criteria	Baseline requirements
WCAG SC 1.4.1 Use of Color : Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.	7. Sensory Characteristics 8. Contrast
WCAG SC 1.3.3 Sensory Characteristics : Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound.	
WCAG SC 1.4.3 Contrast (minimum) : The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following:	

Section 508/WCAG Success Criteria**Baseline
requirements**

-
- Large Text: Large-scale text and images of large-scale text have a contrast ratio of at least 3:1;
 - Incidental: Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement.
 - Logotypes: Text that is part of a logo or brand name has no minimum contrast requirement.

14. Tables

Data Tables

Identify Content

Identify all data tables (including images of data tables) where data cell(s) require header(s) for understanding.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 14.A and 14.B.

Note:

- To assist with identification of data tables, use ANDI:
 1. Launch ANDI: structures, then select the “reading order” link.
 2. The markup added to the page indicates the order that assistive technology will read the content
- Data tables are those tables where information in a cell requires a row or column header to adequately describe the cell's contents. The reading order of a data table will not be sensible when read without the row and/or column headers. The content in a data table would be best understood when it is NOT read in the order that ANDI marks on the page.
- **EXCLUDE** content that does not require a row or column header for understanding:
 - o Layout tables are used for placement of components on the page for visual aesthetics without an informational relationship between headers and the information in the data cells. Content is understandable when read in the marked reading order.
 - o Where content that is visually presented in a table but is understandable when read in the marked reading order, the content does not require a data table structure. This may occur when a CSS technique has been used to visually present information

Check 1.3.1-table-identification

Test Name	Test ID	Test Condition
1.3.1-table-identification	14.A	Each data table has programmatic markup to identify it as a table.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no data tables on the page.

How to Test:

1. Launch ANDI: tables.
 - a. Determine whether ANDI detects and identifies the data table(s). ANDI will identify any non-hidden tables coded using <table>, role="table", or role="grid".
 - i. If the tables module does not display as an option in the modules selection list, then ANDI has not detected any table programmatically on the page (meaning any content presented visually in a table does not have programmatic markup to identify it as a table).
 - ii. If the ANDI: tables module is available, use ANDI's "Analyze Next Table" button to sequentially highlight the detected tables on the page. If the table in question is not outlined by ANDI and/or it is not possible to navigate to the table using ANDI, then ANDI has not detected that table programmatically (meaning the content presented visually in that table does not have programmatic markup to identify it as a table).
2. Review any data tables that use role="presentation".
 - a. ANDI will display role="presentation" in the Element information and/or under Accessibility Alerts.
 - b. A data table that includes role="presentation" will not convey the table semantics to a screen reader and would fail this test.
3. ANDI Output will display an alert whenever ARIA role="table" is not coded correctly.
4. Use the Analyze Previous/Next Table buttons in ANDI to navigate to each of the identified tables on the page.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. It is possible to navigate in ANDI: tables to each data table using the ANDI Analyze Previous/Next Table buttons, AND
2. The data table DOES NOT have an ARIA role="presentation" assigned, AND
3. The data table DOES NOT have any ANDI Table alerts for incorrect use of ARIA table attributes.

Check 1.3.1-cell-header-association

Test Name	Test ID	Test Condition
1.3.1-cell-header-association	14.B	All data cells are programmatically associated with relevant headers.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no data tables on the page.

How to Test:

1. Continue from Test 14.A.
2. Navigate to each data cell with ANDI: tables.
3. Inspect the ANDI Output for each data cell and/or inspect the visual highlighting of the data table to determine whether the table identifies all relevant headers for each data cell.

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. The data table appropriately identifies header relationships for each data cell.

Note:

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

Layout Tables*Identify Content*

Identify any programmatic tables where the table structure is used purely for layout purposes.

EXCLUDE data tables.

Note:

- To find programmatic tables on the page, use ANDI: tables (as in the previous test).
 1. If the ANDI: tables module does not display as an option in the modules selection list, then there is no programmatic table on the page.
- To assist with identifying layout tables:
 1. Launch ANDI: structures, then select the “reading order” link.
 2. The markup added to the page indicates the order that assistive technology will read the content
- Content that is within a layout table does not require row or column headers for understanding. This content should be sensible when read in the reading order identified by ANDI.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 14.C.

Check 1.3.1-layout-table-structure

Test Name	Test ID	Test Condition
1.3.1-layout-table-structure	14.C	The layout table DOES NOT designate the layout table using ARIA role="table" AND DOES NOT include table header structure and relationship elements and/or associated attributes.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there are no layout tables on the page.

How to Test:

1. Continue from Test 14.A.
 - a. If the ANDI: tables module does not display as an option in the modules selection list, then ANDI has not detected a table programmatically on the page, and there is no content on the page that has programmatic markup to identify it as a table.
 - b. If the ANDI: tables module is available, use ANDI's "Analyze Next Table" button to outline the detected tables on the page. If the table in question is not outlined by ANDI and/or it is not possible to navigate to the table using ANDI, then ANDI has not detected that table programmatically (meaning the content presented visually in that table does not have programmatic markup to identify it as a table).
2. Inspect the "Element" output in ANDI to determine whether the layout uses role="table".
3. Inspect the ANDI output and any associated alerts to determine whether a <table> includes header structure elements and/or attributes (e.g., <th>, scope="row").
 - a. If a table has an ARIA role="presentation" assigned and the table also denotes header relationships (e.g., using <th>, scope="row") ANDI will provide a corresponding alert; ignore this alert on a layout table. A table that includes role="presentation" will not convey the table semantics to a screen reader. Therefore, the table structure semantics (e.g., <th>, scope="row") can be ignored if the table is indeed a layout table.

Evaluate Results:

If any of the following is **TRUE**, then the Test Condition is **TRUE** and the content

PASSES:

1. ANDI DOES NOT detect the layout as a table, OR
2. The <table> element includes the attribute role="presentation," OR
3. BOTH of the following are **TRUE**:
 - a. The layout DOES NOT use role="table" or any associated ARIA table attributes (e.g., role="row", role="columnheader"), AND
 - b. The layout DOES NOT include table structure and relationship elements or associated attributes (e.g., <th>, scope="row")

Applicable Standards**Section 508/WCAG Success Criteria****Baseline requirements**

WCAG SC 1.3.1 Info and Relationships: Information, structure, and relationships conveyed through presentation can be programmatically determined. [12. Tables](#)

15. CSS Positioning

CSS Positioning

Identify Content

Use ANDI to identify all content positioned with CSS and inline styles.

1. Launch ANDI, then select the Advanced Settings button; then select “linearize page” to remove CSS positioning from elements on the page.
2. If content is positioned with CSS, the information will be displayed with blue highlighting around the elements and those elements will be placed in the page in the same order in which they appear in the page code.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 15.A.

Check 1.3.2-content-order-meaning-css-position

Test Name	Test ID	Test Condition
1.3.2-content-order-meaning-css-position	15.A	The reading order of the content (in context) is correct and the meaning of the content (in context) is preserved without CSS positioning.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no content positioned using CSS.

How to Test:

1. Review all highlighted, linearized content.
2. Determine whether the reading order of content is still understandable after linearization. If necessary, toggle the linearization button to view the original position of the content. If content becomes illegible due to overlapping, etc., this is not a failure of this test, which is to verify if the programmatic reading order is understandable.

Evaluate Results:

If the following is **TRUE**, then the Test Condition is **TRUE** and the content **PASSES**:

1. The sequence and meaning of the content (in context) is understandable without CSS positioning.

Note:

- ANDI: structures, and the “reading order” link can also be used to reveal the reading order prior to linearizing the content. But it will not identify which content has been positioned with CSS.

Applicable Standards**Section 508/WCAG Success Criteria****Baseline
requirements**

[WCAG SC 1.3.2 Meaningful Sequence](#): When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined.

18. CSS Content and Positioning

16. Pre-Recorded Audio-Only, Video-Only, and Animations

Pre-Recorded Audio-Only

Identify Content

Identify all pre-recorded 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.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 16.A.

Note:

- If audio is *synchronized* with video, slides, animations, or other time-based visual media, use the synchronized media tests in Section 17.

Check 1.2.1-audio-transcript-text

Test Name	Test ID	Test Condition
1.2.1-audio-transcript-text	16.A	A text-based alternative is provided for audio-only content that provides an accurate and complete representation of the audio-only content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no pre-recorded audio-only content.

How to Test:

1. Determine if, for each audio-only content, a transcript is provided.
 - a. Determine whether each transcript is text, (i.e., an image of a transcript would not be sufficient to pass this test).
2. Play the audio-only content entirely while reviewing the transcript.
3. Determine whether the information in the transcript is an accurate, correctly sequenced, and complete representation of the audio-only content.
 - a. To be a complete representation of the content, the transcript must also describe relevant sounds in addition to dialogue, such as doors banging, sirens wailing, identification of speakers in dialogue, etc.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. A text-based transcript is provided for all audio-only content, AND
2. The transcript is an accurate and complete representation of the audio-only content.

Pre-recorded Video-Only

Identify Content

Identify all pre-recorded video-only content.

- In a video-only presentation, information is presented in a variety of ways including animation, text or graphics, the setting and background, the actions and expressions of people, animals, etc.

EXCLUDE any video-only intended as a media alternative for text *if* it is clearly labeled as such.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 16.B.

Note:

- If the video is accompanied by timed sounds or meaningful dialog, it is not video-only; use the *synchronized media* tests in section 17.

Check 1.2.1-video- alternative-equivalent

Test Name	Test ID	Test Condition
1.2.1-video-alternative-equivalent	16.B	The video-only content information is also available through an equivalent text or audio alternative.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no pre-recorded video-only content.

How to Test:

1. Determine whether a text or audio alternative is provided for all video-only content (such as a transcript in text that provides a description of video content and actions or an audio track).

Note: An image of a transcript does not meet this requirement.

2. Play the video-only content entirely while reviewing the text or audio alternative.
3. Determine whether the information in the text or audio alternative includes the same information that the video-only presentation displays (e.g., if the video includes multiple characters, the alternative must identify which character is associated with each depicted action).

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. A text or audio alternative is provided for all video-only content, AND
2. The text or audio alternative is an accurate and complete representation of the video-only content.

Applicable Standards**Section 508/WCAG Success Criteria****Baseline requirements**WCAG SC 1.2.1 Audio-Only and Video-Only (Prerecorded):

For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such:

- Prerecorded Audio-only: An alternative for time-based media is provided that presents equivalent information for prerecorded audio-only content.
- Prerecorded Video-only: Either an alternative for time-based media or an audio track is provided that presents equivalent information for prerecorded video-only content.

16. Audio-Only and Video-Only

17. Synchronized Media

Pre-Recorded Synchronized Media

Identify Content

Identify any pre-recorded synchronized media content.

EXCLUDE media that is clearly identified as a media alternative for text.

If there is no such content, the result for the following test ID(s) is **DOES NOT**

APPLY: 17A and 17.B.

Check 1.2.2-captions-equivalent

Test Name	Test ID	Test Condition
1.2.2-captions-equivalent	17.A	The synchronized media provides accurate captions for the audio content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no pre-recorded synchronized media.

How to Test:

1. Enable captions through the synchronized media player functions and play the media.
 - a. A separate media file with captions may be provided to meet this requirement (i.e. captioned media version is a different file). If provided, test that one.
2. Listen to the audio of the entire synchronized media. Compare the audio to the captions for accuracy, time-synchronization, and equivalence.
 - a. Captions should include all dialogue and equivalents for non-dialogue audio information needed to understand the program content, including sound effects, music, laughter, speaker identification and location.
 - b. The definition of captions includes synchronization. If they are not synchronized, they are not considered captions.
 - c. Captions must not obscure relevant content on the video.

Evaluate Results:

If ALL of the following are **TRUE**, then the Test Condition is **TRUE** and the content

PASSES:

1. Captions are provided for all synchronized media content, AND
2. Captions are accurate and include all dialogue and equivalents for non-dialogue audio information needed to understand the program content, including sound effects, e.g., music, laughter, speaker identification and location, AND
3. All other relevant information in the video is clearly visible (not obstructed by captions) when captions are enabled.

Note:

- Transcripts and non-synchronized alternatives alone will not meet this requirement.

Check 1.2.5-audio-description-equivalent

Test Name	Test ID	Test Condition
1.2.5-audio-description-equivalent	17.B	The synchronized media provides an equivalent soundtrack (combination of narration and audio descriptions) for the video content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no pre-recorded synchronized media.

How to Test:

1. Enable audio descriptions through synchronized media player and play the media.
 - a. Audio descriptions are narration added to or combined with the soundtrack to describe important visual details that cannot be understood from the main soundtrack alone.
 - b. A separate media file with audio description may be provided to meet this requirement (i.e. audio description media file is a different file). If provided, test that one.
2. Identify visual content that requires narrative descriptions.
3. Determine whether the main soundtrack combined with audio descriptions adequately describe important visual details (actions, characters, scene changes, onscreen text, etc.) for a viewer who is unable to see the content.
 - a. If the primary audio adequately describes important visual content in the media, including information about actions, characters, scene changes,

onscreen text, speaker identification and location, and other visual content, additional audio description is not necessary.

4. Compare the video to the combined soundtrack and review the soundtrack for accuracy, time-synchronization, and equivalence.
 - a. Audio descriptions are inserted in pauses in dialog. Synchronization may not be possible, but the description should be provided as timely as possible so meaning is preserved.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The soundtrack (combination of audio descriptions and narration) adequately describes important visual content in the media, including information about actions, characters, scene changes, onscreen text, and other visual content.

Note:

- Transcripts and non-synchronized alternatives alone will not meet this requirement.

Live Synchronized Media

Identify Content

Identify any live synchronized media content. These requirements are only intended for broadcast of synchronized media.

EXCLUDE two-way synchronized media calls between two or more individuals through web apps.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 17.C.

Check 1.2.4-captions-live-equivalent

Test Name	Test ID	Test Condition
1.2.4-captions-live-equivalent	17.C	The live synchronized media provides accurate captions for the audio content.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no live synchronized media.

How to Test:

1. Enable captions through synchronized media player functions.
2. Listen to the audio of the synchronized media. Compare the audio to the captions for accuracy, time-synchronization, and equivalence.
 - a. Lower accuracy of captions for live broadcasts may be acceptable due to limitations of real-time caption capabilities.
 - b. The definition of captions includes synchronization. If they are not synchronized, they are not considered captions.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. Captions are provided for all live synchronized media, AND
2. All captions are accurate, AND
3. Any discrepancies between the captions and the audio output are minor in nature and do not significantly impact understanding (applicable to live captioning only).

Media Player Controls*Identify Content*

Identify any media player used to display synchronized video and audio content.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 17.D to 17.G.

Check 503.4-caption-controls

Test Name	Test ID	Test Condition
503.4-caption-controls	17.D	The media player provides user controls for closed captions.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no media player or if the media player DOES NOT present video synchronized with audio (i.e., it presents audio-only or video-only).

How to Test:

1. Locate the controls for selection of closed captions.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The media player provides user controls for closed captions.

Check 503.4-description-controls

Test Name	Test ID	Test Condition
503.4-description-controls	17.E	The media player provides user controls for audio descriptions.

Applicability

This Test Condition **DOES NOT APPLY (DNA)** if there is no media player or if the media player DOES NOT present video synchronized with audio (i.e., it presents audio-only or video-only).

How to Test:

1. Continue from Test 17.D.
2. Locate the controls for selection of audio descriptions.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The media player provides user controls for audio descriptions.

Media Player - Caption Controls at Volume Menu Level

Identify Content

Identify any media player with volume adjustment and caption controls.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 17.F.

Check 503.4.1-caption-control-level

Test Name	Test ID	Test Condition
503.4.1-caption-control-level	17.F	User controls for captions are provided at the same menu level as the user controls for volume or program selection.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no media player, no caption control, or the media player does not have a volume adjustment control.

How to Test:

1. Continue from Test 17.D.
2. Locate the user controls for volume selection and program selection.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The user controls for captions are provided at the same menu level as the volume controls or program selection controls.

Media Player - Audio Description Controls at Program Menu Level

Identify Content

Identify any media player with program selection and audio description controls.

Note: Program selection is a feature of a media player that allows the user to choose what presentation, or portion of a longer presentation, to play. Program selection is typically the same user experience as opening a file or using a table of contents. The media controls to view an open file (play, pause, fast forward, rewind, etc.) are NOT considered program selection controls.

In most web implementations, media players are typically provided to view specific synchronized media so program selection controls to open any file are not common. Program selection controls to advance to the identified topics in the media (sometimes referred to as “chapters”) may be provided. Volume control is treated as a unique control, distinct from program selection controls.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 17.G.

Check 503.4.2-description-control-level

Test Name	Test ID	Test Condition
503.4.2-description-control-level	17.G	User controls for audio descriptions are provided at the same menu level as the user controls for volume or program selection.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no media player, or the media player does not have a program selection or audio description control.

How to Test:

1. Continue from Test 17.E.
2. Locate the user controls for program selection and volume.

Evaluate Results:

If the following is **TRUE**, then the content **PASSES**:

1. The user controls for audio descriptions are at the same menu level as program selection controls or volume.

Applicable Standards**Section 508/WCAG Success Criteria****Baseline requirements**

[WCAG SC 1.2.2 Captions \(Prerecorded\)](#): Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such.

17. Synchronized Media

[WCAG SC 1.2.3 Audio Description or Media Alternative \(Prerecorded\)](#): An alternative for time-based media or audio description of the prerecorded video content is provided for synchronized media, except when the media is a media alternative for text and is clearly labeled as such.

[WCAG SC 1.2.4 Captions \(Live\)](#): Captions are provided for all live audio content in synchronized media.

[WCAG SC 1.2.5 Audio Description \(Prerecorded\)](#): Audio description is provided for all prerecorded video content in synchronized media.

[Section 508 503.4.1 Caption Controls](#): Where user controls are provided for volume adjustment, ICT shall provide user controls for the selection of captions at the same menu level as the user controls for volume or program selection.

[Section 508 503.4.2 Audio Description Controls](#): Where user controls are provided for program selection, ICT shall provide user controls for the selection of audio descriptions at the same menu level as the user controls for volume or program

Section 508/WCAG Success Criteria**Baseline
requirements**

selection.

18. Resize Text

Textual Content

Identify Content

All text on a page.

EXCLUDE captions for synchronized media and images of text.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 18.A.

Check 1.4.4-resize-text

Test Name	Test ID	Test Condition
1.4.4-resize-text	18.A	There is a mechanism to resize, scale, or zoom in on the text to at least 200% of its original size without loss of content or functionality.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if there is no text on the page.

How to Test:

1. Use built-in browser zoom functions to resize the text to at least 200%.
2. If any of the content did not zoom using the built-in browser functions, determine whether there is a non-AT mechanism to resize page content to 200% of its original size, e.g., Operating System, platform, or other mechanism provided directly by the web page/application.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. There is a non-AT-reliant mechanism that allows the user to resize text to at least 200% of its original size, AND
2. Text is not clipped, truncated or obscured, AND
3. All functionality is available, AND
4. All content is available.

Applicable Standards

Section 508/WCAG Success Criteria

Baseline requirements

[WCAG SC 1.4.4 Resize Text](#): Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality.

[22. Resize Text](#)

19. Multiple Ways

Web Page Access

Identify Content

All web pages within a set of related web pages.

EXCLUDE web pages that are the result of, or a step in, a process, such as an order confirmation form.

If there is no such content, the result for the following test ID(s) is **DOES NOT APPLY**: 19.A.

Check 2.4.5-multiple-ways

Test Name	Test ID	Test Condition
2.4.5-multiple-ways	19.A	There are two or more ways to locate a web page within a set of web pages.

Applicability:

This Test Condition **DOES NOT APPLY (DNA)** if the web page is not within a set of related web pages OR the web page is a result of, or a step in, a process.

How to Test:

1. Determine whether there are two or more ways to locate the specific web page within a set of web pages; these may include (but are not limited to) techniques such as:
 - a. site maps
 - b. site search
 - c. tables of contents
 - d. navigation menus or dropdowns
 - e. navigation trees
 - f. links between pages

Note: Additional techniques for locating a web page may be available beyond those listed in the test instructions.
2. Verify that the identified techniques correctly function and lead to the web page within the site, for example:
 - a. Links/menus lead to the corresponding pages of the site.
 - b. The search form leads to the page(s) which contains the search term.

Evaluate Results:

If ALL of the following are **TRUE**, then the content **PASSES**:

1. At least two techniques exist to locate the web page within the site, AND
2. The techniques function correctly such that they lead to the correct web page.

Applicable Standards

Section 508/WCAG Success Criteria	Baseline requirements
-----------------------------------	-----------------------

WCAG SC 2.4.5 Multiple Ways : More than one way is available to locate a Web page within a set of Web pages except where the Web Page is the result of, or a step in, a process.	23. Multiple Ways
--	-----------------------------------

20. Parsing

Test Name	Test ID	Test Condition
4.1.1 parsing	20.A	This test should be recorded as NOT TESTED.

The result for test 20.A (4.1.1-parsing) should be recorded as **NOT TESTED**.

Note:

- Multiple requirements are specified for the Parsing requirement. To determine if requirements are met, a testing tool would be very helpful but is not available at this time. The test process will be updated when a testing tool is identified. Until then, the test result should be “Not Tested”.

Applicable Standards

Section 508/WCAG Success Criteria

Baseline requirements

[WCAG SC 4.1.1 Parsing](#): In content implemented using markup languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features.

[24. Parsing](#)

Appendix A: Test Process Mapping

Test to Section 508/WCAG Requirement and Baseline Test (cross-reference table)

Test ID / Test Name	Section 508 / WCAG Requirement	Baseline Test
1.A / Alt-version-conformant	Con.1 Conformance Requirement 1. Conformance Level	20. Alternate Versions
1.B / Alt-version-equivalent	Con.1 Conformance Requirement 1. Conformance Level	20. Alternate Versions
1.C / Alt-version-access	Con.1 Conformance Requirement 1. Conformance Level	20. Alternate Versions
1.D / Alt-version-nc-access	Con.1 Conformance Requirement 1. Conformance Level	20. Alternate Versions
1.E / non-interference	Con.5 Conformance Requirement 5. Non-Interference	3. Non-Interference
2.A / 1.4.2-audio-control	1.4.2 Audio Control Con.5 Conformance Requirement 5. Non-Interference	21. Timed Events 3. Non-Interference
2.B / 2.2.2-blinking-moving-scrolling	2.2.2 Pause, Stop, Hide Con.5 Conformance Requirement 5. Non-Interference	21. Timed Events 3. Non-Interference
2.C / 2.2.2-auto-updating	2.2.2 Pause, Stop, Hide Con.5 Conformance Requirement 5. Non-Interference	21. Timed Events 3. Non-Interference
2.D / 4.1.2-change-notify-auto	4.1.2 Name, Role, Value	21. Timed Events
20.A / 4.1.1-parsing	4.1.1 Parsing	24. Parsing
3.A / 2.3.1-flashing	2.3.1 Three Flashes or Below Threshold Con.5 Conformance Requirement 5. Non-Interference	9. Flashing 3. Non-Interference
4.A / 2.1.1-keyboard-access	2.1.1 Keyboard	1. Keyboard Access
4.B / 2.1.1-no-keystroke-timing	2.1.1 Keyboard	1. Keyboard Access

Test ID / Test Name	Section 508 / WCAG Requirement	Baseline Test
4.C / 2.1.2-no-keyboard-trap	2.1.2 No Keyboard Trap Con.5 Conformance Requirement 5. Non-Interference	1. Keyboard Access
4.D / 2.4.7-focus-visible	2.4.7 Focus Visible	2. Focus
4.E / 3.2.1-on-focus	3.2.1 On Focus	2. Focus
4.F / 2.4.3-focus-order-meaning	2.4.3 Focus Order	2. Focus
5.A / 3.3.2-label-provided	3.3.2 Labels or Instructions	10. Forms
5.B / 2.4.6-label-descriptive	2.4.6 Headings and Labels	10. Forms
5.C / 1.1.1-programmatic-label	1.1.1 Non-text Content 4.1.2 Name, Role, Value	10. Forms
5.D / 3.2.2-on-input	3.2.2 On Input	10. Forms
5.E / 4.1.2-change-notify-form	4.1.2 Name, Role, Value	10. Forms
5.F / 3.3.1-error-identification	3.3.1 Error Identification	10. Forms
5.G / 3.3.3-error-suggestion	3.3.3 Error Suggestion	10. Forms
5.H / 3.3.4-error-prevention	3.3.4 Error Prevention (Legal, Financial, Data)	10. Forms
6.A / 2.4.4-link-purpose	2.4.4 Link Purpose (In Context) 4.1.2 Name, Role, Value	14. Links
6.B / 4.1.2-change-notify-links	4.1.2 Name, Role, Value	14. Links
7.A / 1.1.1-meaningful-image-name	1.1.1 Non-text Content 4.1.2 Name, Role, Value	6. Images
7.B / 1.1.1-decorative-image	1.1.1 Non-text Content 4.1.2 Name, Role, Value	6. Images
7.C / 1.1.1- decorative-background-image	1.1.1 Non-text Content 4.1.2 Name, Role, Value	6. Images
7.D / 1.1.1-captcha-alternative	1.1.1 Non-text Content 4.1.2 Name, Role, Value	6. Images
7.E / 1.4.5-image-of-text	1.4.5 Images of Text	6. Images
8.A / 2.2.1-timing-adjustable	2.2.1 Timing Adjustable	21. Timed Events
9.A / 2.4.1-bypass-function	2.4.1 Bypass Blocks	4. Repetitive Content
9.B / 3.2.3-consistent- navigation	3.2.3 Consistent Navigation	4. Repetitive Content

Test ID / Test Name	Section 508 / WCAG Requirement	Baseline Test
9.C / 3.2.4-consistent-identification	3.2.4 Consistent Identification	4. Repetitive Content
10.A / 2.4.6-heading-purpose	2.4.6 Headings and Labels	13. Content Structure
10.B / 1.3.1-heading-determinable	1.3.1 Info and Relationships	13. Content Structure
10.C / 1.3.1-heading-level	1.3.1 Info and Relationships	13. Content Structure
10.D / 1.3.1-list-type	1.3.1 Info and Relationships	13. Content Structure
11.A / 3.1.1-page-language-defined	3.1.1 Language of Page	15. Language
11.B / 3.1.2-part-language-defined	3.1.2 Language of Parts	15. Language
12.A / 2.4.2-page-title-defined	2.4.2 Page Titled	11. Page Titles
12.B / 2.4.2-page-title-purpose	2.4.2 Page Titled	11. Page Titles
12.C / 4.1.2-frame-title	4.1.2 Name, Role, Value	19. Frames and iFrames
12.D / 4.1.2-iframe-name	4.1.2 Name, Role, Value	19. Frames and iFrames
13.A / 1.4.1-color-meaning	1.4.1 Use of Color	7. Sensory Characteristics
13.B / 1.3.3-sensory-info	1.3.3 Sensory Characteristics	7. Sensory Characteristics
13.C / 1.4.3-contrast	1.4.3 Contrast (Minimum)	8. Contrast
14.A / 1.3.1-table-identification	1.3.1 Info and Relationships	12. Tables
14.B / 1.3.1-cell-header-association	1.3.1 Info and Relationships	12. Tables
14.C / 1.3.1-layout-table-structure	1.3.1 Info and Relationships	12. Tables
15.A / 1.3.2-content-order-meaning-CSS-position	1.3.2 Meaningful Sequence	18. Stylesheet Non-dependence
16.A / 1.2.1-audio-transcript-text	1.2.1 Audio-only and Video-only	16. Audio-Only and Video-Only
16.B / 1.2.1-video- alternative-equivalent	1.2.1 Audio-only and Video-only	16. Audio-Only and Video-Only
17.A / 1.2.2-captions-equivalent	1.2.2 Captions (Prerecorded)	17. Synchronized Media

Test ID / Test Name	Section 508 / WCAG Requirement	Baseline Test
17.B / 1.2.5-audio-description-equivalent	1.2.5 Audio Description (Prerecorded)	17. Synchronized Media
17.C / 1.2.4-captions-live-equivalent	1.2.4 Captions (Live)	17. Synchronized Media
17.D / 503.4-caption-description-controls	503.4 User Controls for Captions and Audio Description	17. Synchronized Media
17.E / 503.4.1-caption-control	503.4.1 Caption Controls	17. Synchronized Media
17.F / 503.4.2-description-control	503.4.2 Audio Description Controls	17. Synchronized Media
18.A / 1.4.4-resize-text	1.4.4 Resize text	22. Resize Text
19.A / 2.4.5-multiple-ways	2.4.5 Multiple Ways	23. Multiple Ways

Section 508/WCAG Requirement to Certified Trusted Tester Test and Baseline Test

(cross-reference table)

Section 508 / WCAG Requirement	Test ID / Test Name	Baseline Test
1.1.1 Non-text Content	5.C / 1.1.1-programmatic-label 7.A / 1.1.1-meaningful-image-name 7.B / 1.1.1-decorative-image 7.C / 1.1.1- decorative-background-image 7.D / 1.1.1-captcha-alternative	10. Forms 6. Images
1.2.1 Audio-only and Video-only	16.A / 1.2.1-audio-transcript-text 16.B / 1.2.1-video- alternative-equivalent	16. Audio-Only and Video-Only
1.2.2 Captions (Prerecorded)	17.A / 1.2.2-captions-equivalent	17. Synchronized Media
1.2.4 Captions (Live)	17.C / 1.2.4-captions-live-equivalent	17. Synchronized Media
1.2.5 Audio Description (Prerecorded)	17.B / 1.2.5-audio-description-equivalent	17. Synchronized Media
1.3.1 Info and Relationships	10.B / 1.3.1-heading-determinable 10.C / 1.3.1-heading-level 10.D / 1.3.1-list-type 14.A / 1.3.1-table-identification 14.B / 1.3.1-cell-header-association 14.C / 1.3.1-layout-table-structure	13. Content Structure 12. Tables 18. Stylesheet Non-dependence
1.3.2 Meaningful Sequence	15.A / 1.3.2-content-order-meaning-CSS-position	18. Stylesheet Non-dependence
1.3.3 Sensory Characteristics	13.B / 1.3.3-sensory-info	7. Sensory Characteristics
1.4.1 Use of Color	13.A / 1.4.1-color-meaning	7. Sensory Characteristics
1.4.2 Audio Control	2.A / 1.4.2-audio-control	21. Timed Events
1.4.3 Contrast (Minimum)	13.C / 1.4.3-contrast	8. Contrast
1.4.4 Resize text	18.A / 1.4.4-resize-text	22. Resize Text
1.4.5 Images of Text	7.E / 1.4.5-image-of-text	6. Images
2.1.1 Keyboard	4.A / 2.1.1-keyboard-access 4.B / 2.1.1-no-keystroke-timing	1. Keyboard Access
2.1.2 No Keyboard Trap	4.C / 2.1.2-no-keyboard-trap	1. Keyboard Access
2.2.1 Timing Adjustable	8.A / 2.2.1-timing-adjustable	21. Timed Events

Section 508 / WCAG Requirement	Test ID / Test Name	Baseline Test
2.2.2 Pause, Stop, Hide	2.B / 2.2.2-blinking-moving-scrolling 2.C / 2.2.2-auto-updating	21. Timed Events
2.3.1 Three Flashes or Below Threshold	3.A / 2.3.1-flashing	9. Flashing
2.4.1 Bypass Blocks	9.A / 2.4.1-bypass-function	4. Repetitive Content
2.4.2 Page Titled	12.A / 2.4.2-page-title-defined 12.B / 2.4.2-page-title-purpose	11. Page Titles
2.4.3 Focus Order	4.F / 2.4.3-focus-order-meaning	2. Focus
2.4.4 Link Purpose (In Context)	6.A / 2.4.4-link-purpose	14. Links
2.4.5 Multiple Ways	19.A / 2.4.5-multiple-ways	23. Multiple Ways
2.4.6 Headings and Labels	10.A / 2.4.6-heading-purpose 5.B / 2.4.6-label-descriptive	10. Forms 13. Content Structure
2.4.7 Focus Visible	4.D / 2.4.7-focus-visible	2. Focus
3.1.1 Language of Page	11.A / 3.1.1-page-language-defined	15. Language
3.1.2 Language of Parts	11.B / 3.1.2-part-language-defined	15. Language
3.2.1 On Focus	4.E / 3.2.1-on-focus	2. Focus
3.2.2 On Input	5.D / 3.2.2-on-input	10. Forms
3.2.3 Consistent Navigation	9.B / 3.2.3-consistent-navigation	4. Repetitive Content
3.2.4 Consistent Identification	9.C / 3.2.4-consistent-identification	4. Repetitive Content
3.3.1 Error Identification	5.F / 3.3.1-error-identification	10. Forms
3.3.2 Labels or Instructions	5.A / 3.3.2-label-provided	10. Forms
3.3.3 Error Suggestion	5.G / 3.3.3-error-suggestion	10. Forms
3.3.4 Error Prevention (Legal, Financial, Data)	5.H / 3.3.4-error-prevention	10. Forms
4.1.1 Parsing	20.A / 4.1.1-parsing	24. Parsing

Section 508 / WCAG Requirement	Test ID / Test Name	Baseline Test
4.1.2 Name, Role, Value	12.C / 4.1.2-frame-title 12.D / 4.1.2-iframe-name 2.D / 4.1.2-change-notify-auto 5.C / 1.3.1-programmatic-label 5.E / 4.1.2-change-notify-form 6.A / 2.4.4-link-purpose 6.B / 4.1.2-change-notify-links 7.A / 1.1.1-meaningful-image-name 7.B / 1.1.1-decorative-image 7.C / 1.1.1- decorative-background-image 7.D / 1.1.1-captcha-alternative	19. Frames and iFrames 21. Timed Events 10. Forms 14. Links 6. Images
36 CFR 1194 503.4 User Controls for Captions and Audio Description	17.D / 503.4-caption-description-controls	17. Synchronized Media
36 CFR 1194 503.4.1 Caption Controls	17.E / 503.4.1-caption-control	17. Synchronized Media
36 CFR 1194 503.4.2 Audio Description Controls	17.F / 503.4.2-description-control	17. Synchronized Media
WCAG Conformance Requirement 1. Conformance Level	1.A / Alt-version-conformant 1.B / Alt-version-equivalent 1.C / Alt-version-access 1.D / Alt-version-nc-access	20. Alternate Versions
WCAG Conformance Requirement 5. Non-Interference	1.E / non-interference 2.A/ 1.4.2-audio-control 2.B/ 2.2.2-blinking-moving-scrolling 2.C/ 2.2.2-auto-updating 3.A/ 2.3.1-flashing 4.C/ 2.1.2-no-keyboard-trap	3. Non-Interference

Appendix B: Document Change Log

Note: Minor punctuation, formatting and spelling changes not included.

Version 5.1.1, August 2022

Version 5.1.1 makes some clarifications to the test process and supporting content from Version 5.1. There are a few significant changes to the test process:

- Added Appendix D for possible ANDI workarounds.
- 1.A to 1.C: Conforming alternate versions should be identified as such. Failure of a test in this section results in FAIL rather than DNA so that testing continues. Test for “one path” to conforming version removed as this may be impossible to verify.
- 1.C removes reference access from non-conforming version.
- 5.A and 5.B: Clarification that these tests are for visual labels.
- 6.A: Sentences and list items are acceptable link context.
- 7.A: Meaningful images, including images that suggest mood or tone, should be identified even if they are described in nearby text. This allows AT users to be aware of their presence and location.
- 10.D: Exclude menus and navigational framework.
- 13.A: Clarification to check if color is only visual method used.
- 13.B: Color added as a type of sensory information and therefore cannot be used in combination with another type of sensory information to pass this test.
- 15.A: Not testing for overlapping or other illegibility issues.

Please see the Version 5.1 section for summary of changes from Version 5.0.

Location in 5.1	Change
Testing Tools	Added reference to App. D for ANDI workarounds
Conforming Alternate Version and Non-Interference	<p>General</p> <ul style="list-style-type: none"> • Introduction revised; some headings renamed <p>Changes to testing process</p> <ul style="list-style-type: none"> • Failures of 1.A through 1.C now marked as FAIL instead of DNA to allow for continued testing of version identified as the alternate version. • “Alternate” version changed to “identified” version • 1.C removes reference to access from non-conforming version • Removed test 1.D. Simplified the mechanism test to no longer check for path taken (whether from conforming or non-conforming version, etc.) • Conforming alternate versions must be identified, e.g., through a label on the page, as a link, in documentation, user preferences, controls to modify text appearance, etc.
4.C	How to Test, Step 1: Changed “Tab through...” to “Use standard navigation keys...”
5.A	Clarified that this test addresses presence of visual labels,

Location in 5.1	Change
	and that accuracy of visual labels is not evaluated in this test
5.B	<p>General</p> <ul style="list-style-type: none"> Clarified that this test is for visual labels only <p>Evaluate Results</p> <ul style="list-style-type: none"> Added “OR” after condition #3
5.C	5.C now references SC 1.1.1 instead of 1.3.1
5.E	Removed Note item about testing revealed content per 4.G
5	Applicable standards: added SC 1.1.1 to replace SC 1.3.1
6.A	How to Test, Step 2: added “sentence” and clarified “list item” as acceptable link context
7	<p>This test has been updated so that most images that convey information should be considered meaningful. This allows AT users to be aware of their presence and location.</p> <p>Identify Content</p> <ul style="list-style-type: none"> First bullet: Added “All”, removed “important” before “meaningful images” – now “All images which convey information should be considered meaningful images” Last bullet added: “Images are not pure decoration if they serve to illustrate text content, suggest mood or tone, or create specific sensory experiences, such as a section dividers or bullet icons that use imagery or designs that convey specific moods or tones. If they convey any information, AT users should receive the same benefit of that information.” Added to Notes: <ul style="list-style-type: none"> “WCAG defines “pure decoration” as “serving only an aesthetic purpose, providing no information, and having no functionality.” WCAG defines “specific sensory experiences” as “a sensory experience that is not purely decorative and does not primarily convey important information or perform a function.”
7.A	Note: changed “These cannot be set” to “These should not be set”
10.D	<p>Identify Content</p> <ul style="list-style-type: none"> Replaced note to clarify that navigation framework elements such as menus should be excluded: “Exclude menus or other elements that are part of the page’s navigational framework from this test. Developers might use list elements to create grouped navigational items, such as menus, submenus, lists of navigation links, breadcrumbs, etc. while styling them to remove bullets or numbering and orienting the lists horizontally instead of vertically. If these navigational elements are

Location in 5.1	Change
	<p>part of the page's navigational framework, separate from the main content, they should not be considered visually apparent lists for this test and should be excluded. However, any lists which are part of the main content should be included."</p> <p>How to Test</p> <ul style="list-style-type: none"> Step 3: Added "Exclude menus and navigational framework elements from this test, unless they are part of the main content."
12.D	Note: added note "If ANDI fails to run in an iframe, see Appendix D for possible workarounds."
13.A	How to Test: Added "visual" to "Determine whether color is the only visual method used to convey information"
13.B	<p>Identify Content, Test Condition, How to Test, Evaluate Results</p> <ul style="list-style-type: none"> Added color as a type of sensory information <p>Removed note "The use of color can be used in combination with ... to meet this requirement", etc.</p>
14.A	<p>How to Test</p> <ul style="list-style-type: none"> Step 1.a: added "ANDI will identify any non-hidden tables coded using <table>, role="table", or role="grid"."
15.A	<p>How to Test</p> <ul style="list-style-type: none"> Step 2: added "If content becomes illegible due to overlapping, etc., this is not a failure of this test, which is to verify if the programmatic reading order is understandable."
17.C	Identify Content: "If there is no such content, the result for the following test ID(s) is DOES NOT APPLY: 17.D to 17.G" instead of "to 17.F"
17.E	Identify Content: "If there is no such content, the result for the following test ID(s) is DOES NOT APPLY: 17.F" instead of 17.E
17.F	Test Name is now "503.4.1-caption-control-level"
17.G	<ul style="list-style-type: none"> Identify Content: "If there is no such content, the result for the following test ID(s) is DOES NOT APPLY: 17.G" instead of 17.F Test name is now "503.4.2-description-control-level"

Version 5.1, January 2021

This section is included as a reference for testers who were using Version 5.0. Version 5.1 introduced some significant changes from Version 5.0:

- Buttons testing moved from 6: Links/Buttons to Forms. Topic 6 now only tests links.

- 5.C checks for other programmatic associations.
- 5.E test includes button names.
- 6.A does not apply to anchors or hidden links.
- Images not considered decorative due to nearby text description.
- 10.B compares ANDI output and visual heading.
- 12.D provides more information on handling negative tabindex.
- 13.B allows use of “above” and “below” to reference sequence.
- 15.A test for ::before and ::after removed.
- Reorganization and clarification of audio and caption control tests.

Location in 5.0	Change
various	Update Baseline Test Numbering: <ul style="list-style-type: none"> • The name of 2. Focus Visible changed to 2. Focus. • The new 2. Focus now also includes what was formerly in 3. Focus Order. • The numbering for 25. Non-Interference changed to 3. Non-Interference.
Page 5	Added to Conformance Reporting requirements: If a tester cannot complete a test, a note should be added to the test report indicating “This test could not be performed”, with a detailed explanation of the issue.
Page 12	1.E, non-interference added note to clarify Flashing result of Not Tested does not pass the evaluation result requirements.
Page 15	2.A, 1.4.2-audio-control <ul style="list-style-type: none"> • added bullet to “Notes”: The Certified Trusted Tester process requires the mechanism within three elements for clear measurability. This requirement is not specified in WCAG. To meet the non-interference requirement, the mechanism can be a focusable element or text instructions at the top of the page prior to repetitive content.
Page 16	2.B, 2.2.2-blinking-moving-scrolling <ul style="list-style-type: none"> • Added “Note” to the Identify Content section to clarify when 2.B, 2.C, or both apply • Moved text (including scrolling text, videos, and synchronized media) to beneath first bullet and changed order of items in the list for clarity; “scrolling text” is now the last item in the list
Page 17	2.C, 2.2.2-auto-updating <ul style="list-style-type: none"> • Added “Note” to the Identify Content section to clarify when 2.B, 2.C, or both apply • Changed “moving/blinking/scrolling” to “auto-updating” to now read “three elements before/after the auto-updating content” in Evaluate Results section.

Location in 5.0	Change
Page 21	4.A, 2.1.1-keyboard-access How to Test instructions broadened to include additional methods of identifying functionality.
Page 22	4.A, 2.1.1-keyboard-access Note added to identify common methods of documenting shortcut keys.
Page 25	<p>4.F, 2.4.3-focus-order-meaning</p> <ul style="list-style-type: none"> Added directions to incorporate revealed content into the test steps for focus-order-meaning. (This eliminates a separate test result for 4.G.) “It may be necessary to use the keyboard to activate trigger controls that reveal hidden content with focusable elements (e.g., menus, dialogs, modal dialog boxes, expandable tree list) to check the focus order to, from, and within the revealed content.” Clarified that focus must also remain in modal dialog boxes when navigating backwards. Added bulleted content to the “Notes”: <ul style="list-style-type: none"> Focus order does not necessarily need to be top to bottom, left to right. When the focus order does not affect meaning or operability, this test Does Not Apply. Example: A row of icons linking to social media may not need to be navigated in a particular order.
Page 25	4.G, 2.4.3-focus-order-reveal as a separate test condition and ID was removed for redundancy. A test step was added to 4.F
Page 26	4.H, 2.4.3-focus-order-return as a separate test condition and ID was removed for redundancy. This condition is already covered in 4.F steps to check backward focus for meaning and operability.
Page 27	<p>Topic 5: In the Identified Content section, added “focusable element” to clarify the ANDI module used to identify form elements.</p> <p>Buttons testing was uncoupled from Topic 6 (Links/Buttons) and included as part of Forms testing.</p> <ul style="list-style-type: none"> “Identifying Form Components” added “buttons” as an example of a form element to be tested in this topic. Terminology changed from “form field” to “form elements” or “form components/controls”.

Location in 5.0	Change
Page 28	<p>Test 5.B, 2.4.6-label-descriptive added Evaluation result: 2. Each button label is sufficiently clear and descriptive, so users know its function.</p> <p>Modified test step #4 to 5.C, 1.3.1-programmatic-label to clarify other programmatic associations are reviewed “If the ANDI Output does not adequately define the form element,” and added sub-step a. This language was previous part of the test language in Topic 6 for buttons.</p> <p>a. In cases where the purpose of the button is intentionally vague or ambiguous (e.g., the content to be revealed after selecting a link to “Door 1,” “Door 2,” or “Door 3” is intended to be a surprise), it may be sufficient for the combination of button text, accessible name, accessible description, and/or button context to refer to the button purpose vaguely or ambiguously.</p> <p>Evaluation result # 4 was added to 5.C, 1.3.1-programmatic-label. “The combination of the programmatically determined button context and the ANDI Output provide adequate description of each buttons’ purpose.”</p>
Page 29	<p>Test 5.E, 4.1.2-change-notify-form now specifically includes “button names” as a possible form element to be tested for interactions that trigger changes.</p>
Page 31	<p>Test 5.F, 3.3.1-error-identification clarifies the test does not apply if the “form element” (rather than the page) does not have automatic error detection.</p>
Page 34 & 35	<p>Topic 6 tests links alone. Buttons are now tested as part of Forms (Topic 5). The test condition for 6.A, 2.4.4-link-purpose and 6.B, 4.1.2-change-notify-links no longer include references to buttons.</p> <p>Changes to “Notes” for 6.A, 2.4.4-link-purpose</p> <ul style="list-style-type: none"> Added: This test does not apply to links that function as an anchor or target and are not perceivable or selectable by users. Removed: This test also covers the requirement for WCAG SC 4.1.2 Name, Role, Value.

Location in 5.0	Change
Page 36	<p>A change to our understanding of Decorative Images removes the ability to label an image “decorative” when text provides the same information.</p> <ul style="list-style-type: none"> • 7.A, 1.1.1-meaningful-image-name no longer states: “If the ANDI Output points to page content for the image’s description, determine whether the description is provided.” • Instead the Test Step 1.a instructs: “The ANDI Output must provide an equivalent description of the image. This description can be provided by a brief description of the image in the ANDI Output and instructions about where to obtain the full description.”
Page 42	<p>The order of test steps was modified for 9.A, 2.4.1-bypass-function to first check for keyboard-accessible methods to bypass repetitive content before launching ANDI: focusable elements.</p>
Page 47	<p>10.B, 1.3.1-heading-determinable added in How to Test, step 3 “Review the ANDI Output for each heading to determine if it matches the visual heading. If they do not match, then the heading is not properly defined programmatically.”</p>
Page 55	<p>Instructions for How to Test, step 3 of 12.C, 4.1.2-frame-title changed to “Navigate back to the page being tested and launch ANDI again” (from “Use the browser’s Back button to return to the page being tested and launch ANDI again”). Change was made due to different navigation behaviors of various browsers.</p>
Page 56	<p>Changes to 12.D, 4.1.2-iframe-name to clarify how to address iframes with a negative tabindex.</p> <ul style="list-style-type: none"> • Created a note in the Identify Content which moved some content and added further explanation regarding excluding iframes when they have a negative tabindex. • Clarified test applicability to account for browser inconsistency’s in handling the tab order of iframes. New wording states: “This Test Condition DOES NOT APPLY (DNA) if no iframes are identified or if the iframes are explicitly excluded from the tab order.” • Added text “or where the tabindex is not defined” to test step 2 to now read, “Review the ANDI Output for each iframe that has a non-negative tabindex value or where the tabindex is not defined to determine whether the accessible name and description accurately describe the content of each <iframe>.”

Location in 5.0	Change
Page 59	<p>Addition to 13.B, 1.3.3-sensory-info</p> <p>Note: The use of “above” or “below” is allowable when it references the sequential order of content.</p>
Page 66	<ol style="list-style-type: none"> 1. Removed CSS Content subsection which had Test ID 15.A, 1.3.1-meaningful-content-css-before-after --“For the meaningful content provided via CSS pseudo-elements ::before and ::after, equivalent information is available in another way.” Newer Assistive Technology accounts for content using CSS pseudo-elements ::before and ::after. The ANDI computation has already been updated 2. Removed WCAG SC 1.3.1 Info and Relationships from applicable Standards for this section. 3. Renumbered 15.B. It is now: 15.A, 1.3.2-content-order-meaning-css-position
Page 73	<p>Removed audio-description control mechanism from Test ID 17.D.</p> <ul style="list-style-type: none"> • The Test Name changed to 503.4-caption-controls (from 503.4-caption-description-controls) • Text “and audio descriptions” was removed from the Test Condition, Test Steps, and Evaluate Results. • The Test Condition is now: The media player provides user controls for closed captions. <p>Note: This change was made to provide better clarity in tests for the menu level of the mechanism</p>
Page 74	<p>Added a separate test for audio-description control mechanism to create new Test ID 17.E, 503.4-audio-controls.</p> <p>Note: This change was made to provide better clarity in tests for the menu level of the mechanism.</p>
Page 74	<p>Renumbered Test ID 17.E to 17.F, 503.4.1-caption-control.</p> <ul style="list-style-type: none"> • Added “and caption controls” to Identify Content to now read, “Identify any media player with volume adjustment and caption controls.” • Added “no caption control” to Applicability: This Test Condition DOES NOT APPLY (DNA) if there is no media player, no caption control, or the media player does not have a volume adjustment control.

Location in 5.0	Change
Page 75	<p>Renumbered Test ID 17.F to 17.G, 503.4.2-description-control</p> <ul style="list-style-type: none">• Added “and audio description controls” to Identify Content: Identify any media player with program selection and audio description controls.• Added “and audio description” to Applicability: This Test Condition DOES NOT APPLY (DNA) if there is no media player, or the media player does not have a program selection and audio description control.

Appendix C: Test Process Quick Reference

Quick Reference with Test Conditions

Test ID	Test Name	Test Condition
1.A	alt-version-conformant	An alternate version passes all applicable Test Conditions in this test process.
1.B	alt-version-equivalent	The accessible version is up-to-date with the same information and functionality.
1.C	alt-version-access	The mechanism to reach the accessible equivalent version from the non-conforming page is accessible.
1.D	alt-version-nc-access	The non-conforming version(s) can only be reached from conforming content.
1.E	non-interference	Content in the non-conforming version(s) does not interfere with the user's ability to access or use the conforming content.
2.A	1.4.2-audio-control	The user can pause, stop, or control the volume of audio content that plays automatically.
2.B	2.2.2-blinking-moving-scrolling	The user can pause, stop, or hide moving, blinking, or scrolling content.
2.C	2.2.2-auto-updating	The user can pause, stop, hide, or control the frequency of automatically updating content.
2.D	4.1.2-change-notify-auto	The page provides notification of each automatic update/change in content.
3.A	2.3.1-flashing	If NO flashing content is found, then this Test Condition DOES NOT APPLY (DNA). If flashing content IS found, then this test should be recorded as NOT TESTED.
4.A	2.1.1-keyboard-access	All functionality can be accessed and executed using only the keyboard.
4.B	2.1.1-no-keystroke-timing	Individual keystrokes do not require specific timings for activation of functionality.
4.C	2.1.2-no-keyboard-trap	There is no keyboard trap.
4.D	2.4.7-focus-visible	A visible indication of focus is provided when focus is on the interface component.
4.E	3.2.1-on-focus	When an interface component receives focus, it does not initiate an unexpected change of context.
4.F	2.4.3-focus-order-meaning	The focus order preserves the meaning and operability of the web page.
5.A	3.3.2-label-provided	Labels and instructions are provided for each form input.

Test ID	Test Name	Test Condition
5.B	2.4.6-label-descriptive	Each form label is sufficiently descriptive.
5.C	1.3.1-programmatic-label	The combination of the accessible name, accessible description, and other programmatic associations (e.g., table column and/or row associations) describes each input field and includes all relevant instructions and cues (textual and graphical).
5.D	3.2.2-on-input	Changing field values/selections (e.g., entering data in a text field, changing a radio button selection) does NOT initiate an unexpected change of context.
5.E	4.1.2-change-notify-form	The page provides notification of each form-related change in content.
5.F	3.3.1-error-identification	The item in error is identified and the error is described to the user in text.
5.G	3.3.3-error-suggestion	Additional guidance (e.g., suggestion for corrected input) is provided about how to correct errors for form fields.
5.H	3.3.4-error-prevention	The web page allows the user to check, reverse, and/or confirm submission.
6.A	2.4.4-link-purpose	The purpose of each link or button can be determined from any combination of the link/button text, accessible name, accessible description, and/or programmatically determined link/button context.
6.B	4.1.2-change-notify-links	The page provides notification of each change in content that is the result of interaction with a link or button.
7.A	1.1.1-meaningful-image-name	The accessible name and accessible description for a meaningful image provides an equivalent description of the image.
7.B	1.1.1-decorative-image	There is no accessible name and accessible description for a decorative image.
7.C	1.1.1- decorative-background-image	The background image is not the only means used to convey important information.
7.D	1.1.1-captcha-alternative	Alternative forms of CAPTCHA are provided.
7.E	1.4.5-image-of-text	The image of text cannot be replaced by text or is customizable.
8.A	2.2.1-timing-adjustable	The user can turn off, adjust, or extend the time limit.
9.A	2.4.1-bypass-function	A keyboard-accessible method is provided to

Test ID	Test Name	Test Condition
		bypass repetitive content.
9.B	3.2.3-consistent-navigation	Each navigational element occurs in the same relative order with regard to other repeated components on each web page where it appears.
9.C	3.2.4-consistent-identification	The accessible name and description is consistent for components that perform the same function.
10.A	2.4.6-heading-purpose	Each heading describes the topic or purpose of its content.
10.B	1.3.1-heading-determinable	Each programmatically determinable heading is a visual heading and each visual heading is programmatically determinable.
10.C	1.3.1-heading-level	Programmatic heading levels logically match the visual heading presentation within the heading structure.
10.D	1.3.1-list-type	All visually apparent lists are programmatically identified according to their type.
11.A	3.1.1-page-language-defined	The default human language of each web page can be programmatically determined.
11.B	3.1.2-part-language-defined	The human language for any content segment that differs from the default human language of the page can be programmatically determined.
12.A	2.4.2-page-title-defined	A <title> element is defined for the web page.
12.B	2.4.2-page-title-purpose	The <title> element identifies the contents or purpose of the web page.
12.C	4.1.2-frame-title	Each <frame> has a title attribute that describes its content.
12.D	4.1.2-iframe-name	The combination of accessible name and description for each <iframe> describes its content.
13.A	1.4.1-color-meaning	Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.
13.B	1.3.3-sensory-info	Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components, such as shape, size, visual location, orientation, or sound.
13.C	1.4.3-contrast	The visual presentation of text and images of text have sufficient contrast.
14.A	1.3.1-table-identification	Each data table has programmatic markup to identify it as a table.
14.B	1.3.1-cell-header-	All data cells are programmatically associated with

Test ID	Test Name	Test Condition
	association	relevant headers.
14.C	1.3.1-layout-table-structure	The layout table DOES NOT designate the layout table using ARIA role="table" AND DOES NOT include table header structure and relationship elements and/or associated attributes.
15.A	1.3.2-content-order-meaning-CSS-position	The reading order of the content (in context) is correct and the meaning of the content (in context) is preserved without CSS positioning.
16.A	1.2.1-audio-transcript-text	A text-based alternative is provided for audio-only content that provides an accurate and complete representation of the audio-only content.
16.B	1.2.1-video-alternative-equivalent	The video-only content information is also available through an equivalent text or audio alternative.
17.A	1.2.2-captions-equivalent	The synchronized media provides accurate captions for the audio content.
17.B	1.2.5-audio-description-equivalent	The synchronized media provides an equivalent soundtrack (combination of narration and audio descriptions) for the video content.
17.C	1.2.4-captions-live-equivalent	The live synchronized media provides accurate captions for the audio content.
17.D	503.4-caption-description-controls	The media player provides user controls for closed captions and audio descriptions.
17.E	503.4.1-caption-control	User controls for captions are provided at the same menu level as the user controls for volume or program selection.
17.F	503.4.2-description-control	User controls for audio descriptions are provided at the same menu level as the user controls for volume or program selection.
18.A	1.4.4-resize-text	There is a mechanism to resize, scale, or zoom in on the text to at least 200% of its original size without loss of content or functionality.
19.A	2.4.5-multiple-ways	There are two or more ways to locate a web page within a set of web pages.
20.A	4.1.1-parsing	This test should be recorded as NOT TESTED.

One-Page Quick Reference – Test Names Only

Test ID	Test Name
1.A	alt-version-conformant
1.B	alt-version-equivalent
1.C	alt-version-access
1.D	alt-version-nc-access
1.E	non-interference
2.A	1.4.2-audio-control
2.B	2.2.2-blinking-moving-scrolling
2.C	2.2.2-auto-updating
2.D	4.1.2-change-notify-auto
3.A	2.3.1-flashing
4.A	2.1.1-keyboard-access
4.B	2.1.1-no-keystroke-timing
4.C	2.1.2-no-keyboard-trap
4.D	2.4.7-focus-visible
4.E	3.2.1-on-focus
4.F	2.4.3-focus-order-meaning
5.A	3.3.2-label-provided
5.B	2.4.6-label-descriptive
5.C	1.3.1-programmatic-label
5.D	3.2.2-on-input

Test ID	Test Name
5.E	4.1.2-change-notify-form
5.F	3.3.1-error-identification
5.G	3.3.3-error-suggestion
5.H	3.3.4-error-prevention
6.A	2.4.4-link-purpose
6.B	4.1.2-change-notify-links
7.A	1.1.1-meaningful-image-name
7.B	1.1.1-decorative-image
7.C	1.1.1- decorative-background-image
7.D	1.1.1-captcha-alternative
7.E	1.4.5-image-of-text
8.A	2.2.1-timing-adjustable
9.A	2.4.1-bypass-function
9.B	3.2.3-consistent-navigation
9.C	3.2.4-consistent-identification
10.A	2.4.6-heading-purpose
10.B	1.3.1-heading-determinable
10.C	1.3.1-heading-level
10.D	1.3.1-list-type

11.A	3.1.1-page-language-defined
11.B	3.1.2-part-language-defined
12.A	2.4.2-page-title-defined
12.B	2.4.2-page-title-purpose
12.C	4.1.2-frame-title
12.D	4.1.2-iframe-name
13.A	1.4.1-color-meaning
13.B	1.3.3-sensory-info
13.C	1.4.3-contrast
14.A	1.3.1-table-identification
14.B	1.3.1-cell-header-association
14.C	1.3.1-layout-table-structure
15.	1.3.2-content-order-meaning-CSS-position
16.A	1.2.1-audio-transcript-text
16.B	1.2.1-video-alternative-equivalent
17.A	1.2.2-captions-equivalent
17.B	1.2.5-audio-description-equivalent
17.C	1.2.4-captions-live-equivalent
17.D	503.4-caption-description-controls
17.E	503.4.1-caption-control

17.F	503.4.2- description-control
18.A	1.4.4-resize-text

19.A	2.4.5-multiple- ways
20.A	4.1.1-parsing

Appendix D: ANDI Workarounds

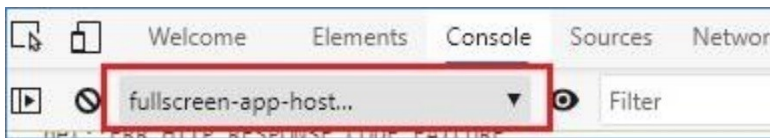
On some web pages, ANDI might not run as expected. The following information is provided to diagnose the problem and possibly allow ANDI to run on these pages.

iFrame Workaround

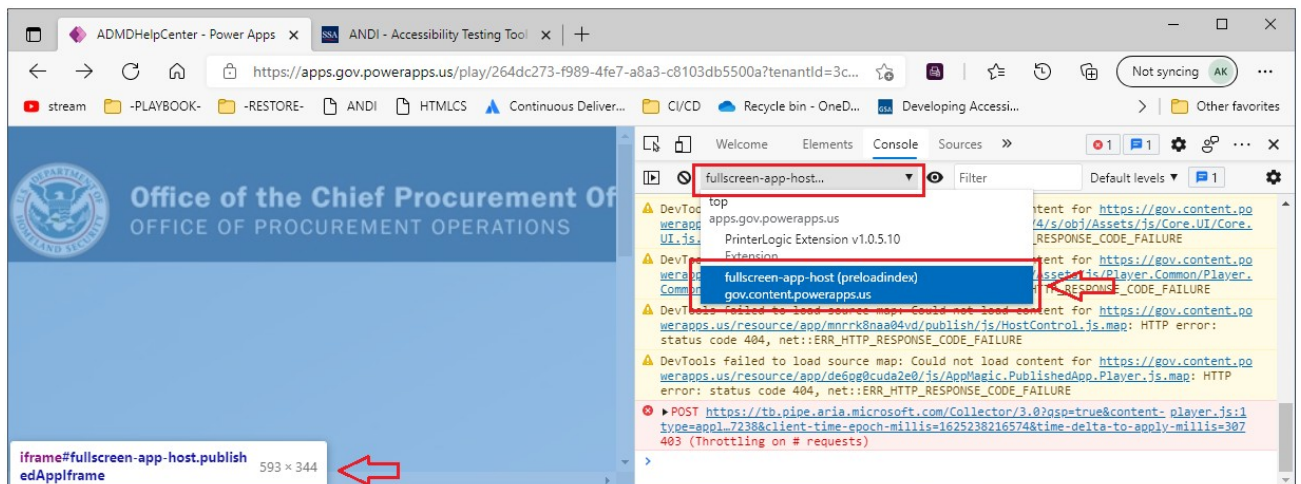
If ANDI will not run in an iframe, a possible workaround is as follows. (The following instructions are for Microsoft Edge; if necessary, try this using the developer tools in other browsers.)

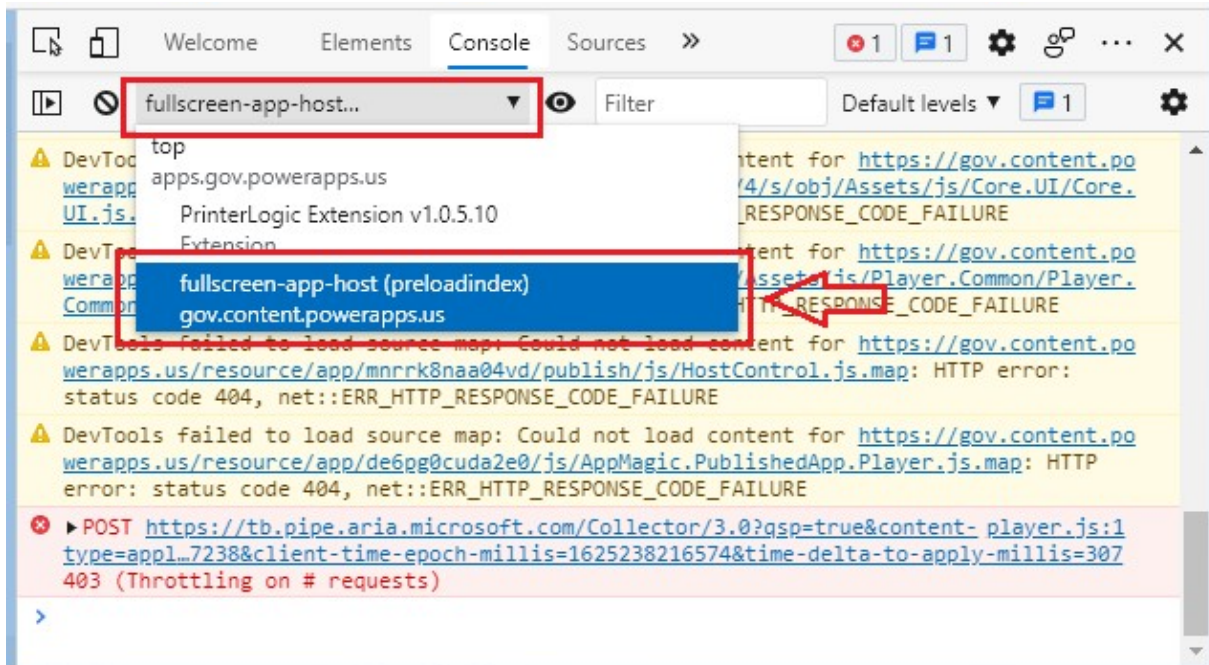
1. Right click on the content contained by the iframe. Select Inspect to open DevTools.
2. Select the console tab within DevTools.
3. Access the “JavaScript Context” dropdown in DevTools to confirm you are inside the correct iFrame container as follows:

- a. Select the JavaScript Context dropdown:

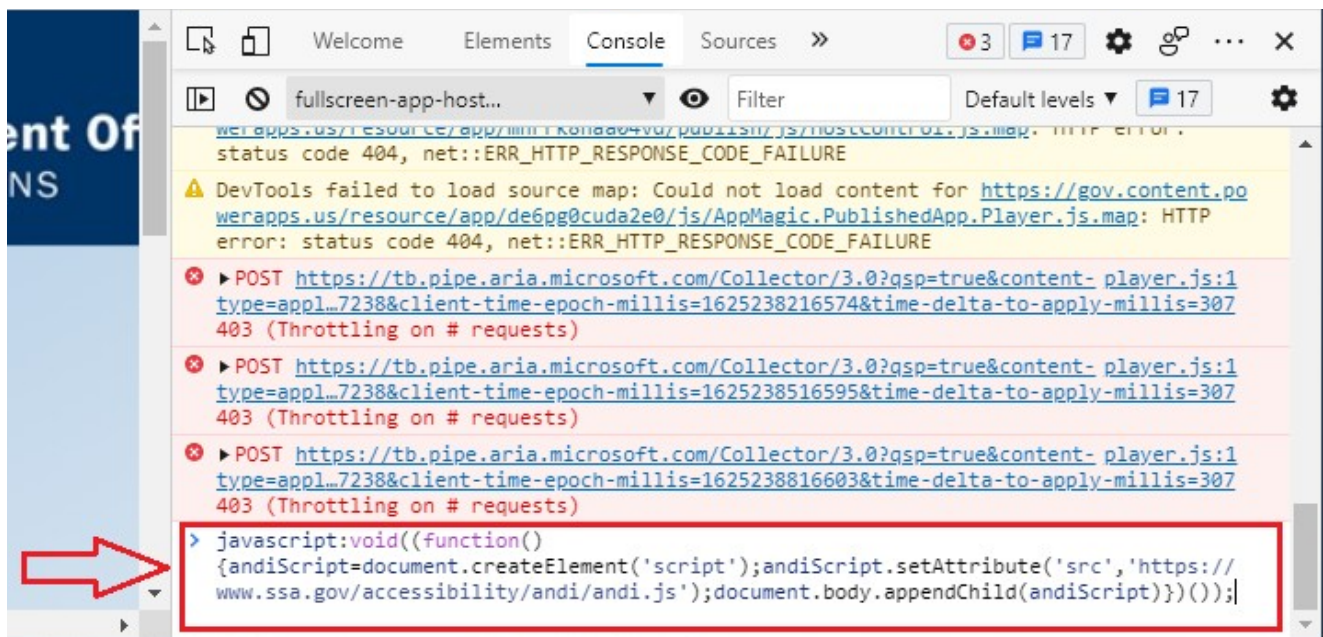


- b. Open the JavaScript Context dropdown and hover the mouse pointer over entries in the dropdown until the UI on the left of the DevTools screen highlights. When the UI on the left highlights, **select that item in the dropdown**. The highlighting is confirmation that the iFrame hovered over in the JavaScript context dropdown is the iFrame container in which you should paste the ANDI code:





- Copy the ANDI favelet code from the browser and paste into the iFrame container at the ">" prompt at the bottom of the DevTools Console. Select "ENTER" to start the ANDI tool within the iFrame container:



- ANDI should run normally within the iFrame container.

Other Issues

Some sites have content security policy (CSP) enhancements to prevent outside scripts from executing on the page to prevent phishing and other security threats. There may also be scripting restrictions. These may interfere with ANDI running on these pages.

SSA provides possible workarounds under ANDI's FAQ in the "Why won't ANDI work on this page?" section at

<https://www.ssa.gov/accessibility/andi/help/faq.html#wontLaunch>.

Note: Check with your IT Department to obtain approval before trying these workarounds. Be sure to re-enable CSP and other restrictions after testing.