Trusted Tester Section 508 Compliance Test Process for Applications

April 2017 | Version 4.0

Contents

About this document

Who should use this document

This document has been designed for and is intended for use by Trusted Testers.

A Trusted Tester is a person certified to provide accurate and repeatable Section 508 compliance test results for software and Web applications. A Trusted Tester follows the Section 508 compliance Application Test Process, uses approved testing tools, and evaluates software and web applications for compliance with Section 508 standards. Trusted Testers are those who have taken the Trusted Tester Training Course and passed the Certification Exam.

For more information on the DHS Trusted Tester Training Course and Exam, contact the DHS Accessibility Helpdesk at accessibility@dhs.gov.

DHS Test Process and Harmonized Testing

The Department of Homeland Security (DHS), Office of Accessible Systems & Technology (OAST) has a mission to provide strategic direction, technical support, and training to ensure agency employees and customers with disabilities have equal access to information and data.

Section 508 of the Rehabilitation Act of 1973, as amended (29 U.S.C. 794d), requires all federal departments and agencies to ensure that their electronic information & technology (EIT) is accessible to people with disabilities. This Section 508 Compliance Test Process for Applications has been produced in support of this mission.

The document incorporates the "Harmonized Processes for Section 508 Testing: Baseline Tests for Software & Web Accessibility" version 2.0.2. The baseline tests established the minimum steps required to determine compliance with Section 508 requirements. A unified test approach for Section 508 compliance will provide consistency across government and confidence in test results shared between agencies.

Test instructions that are specific to DHS only are identified with *DHS-specific* or "[no baseline]." The outcomes of these tests will be reflected only in Section 508 test results. Baseline test results are not affected by DHS-specific tests.

How This Document is Structured

Section 1 describes the **rationale** for combining Software and Web testing into one test process. The rationale behind each individual test is also provided.

Section 2 provides details of the **test environment for this process (OS, browser, testing tools)**, including installation and settings guidance. Additionally, a decision chart provides guidance on the circumstances for determining appropriate testing tools and applicable standards.

Section 3 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 evaluate compliant or not compliant for test conditions.

Section 4 provides **cross-references tables** to indicate the relationship of the Tests to Section 508 requirements and Baseline tests.

Section 1: Introduction and Rationale for Tests

Applying the Section 508 Standards and Baseline Tests to the Applications Test Process

The Section 508 technical standards include one section aimed at browser-based information (Web), and another section aimed at native applications and operating systems (Software). Applications often incorporate interface elements where both Software and Web standards apply. It is rare to find an application that has only Web interface elements or only Software interface elements. For example, a Software Application might include a Web-based help page, and a Web Application page might include Flash (software) for a video clip display. Since the majority of applications contain both Web and software interface elements, testing for both Software and Web standards are combined into one process.

There are twelve technical Software standards (a through l) and sixteen Web standards (a through p). Some compliance requirements, such as color non-dependence, flicker restrictions, and form labeling, appear under both the software and Web technical standards.

In addition to the technical requirements for Software and Web, the Section 508 Standards contain a section on Functional Performance Criteria (FPC). The FPC require that "at least one mode of operation and information retrieval that does not require user [ability] shall be provided, or support for assistive technology used" by people who are blind, have low vision, no hearing, difficulty hearing, speech disabilities, and limited physical capabilities.

The harmonized Baseline tests were developed to contain sufficient information and instruction to make a consistent and unambiguous assessment of the accessibility of interface components, independently of the other tests. Test outcomes are provided for Section 508 standards, WCAG 2.0, and the baseline test.

Through experience, it was found that testing the Standards in the alphabetical order of the 508 technical requirements was inefficient and resulted in needless repetitive testing. The test process begins with evaluating elements that are most commonly found in applications and therefore most likely to have applicable accessibility requirements. Some test steps apply only to Software while others only to Web elements. There are 15 main test processes; some have multiple tests within the process and some apply only to software or only to Web. Steps 10 through 15 apply only to Web.

The test process is therefore designed to streamline the sequence for testers. Version 2.8 of the Test Process provided results for only the Section 508 Software and Web technical standards. This updated Test Process provides test outcomes for both the Section 508 (technical and functional performance) requirements and the Baseline tests.

Other 508 Standards and Other Types of Products

The Applications test process herein was developed to test applications running on the Microsoft Windows Operating System (used at DHS) and in Microsoft Internet Explorer (the default browser used at DHS) on a system with a physical keyboard. Although Section 508 Standards also include “Telecommunications products,” “Self-contained, closed products,” and “Desktop and portable computers,” these typically do not apply to these applications and are not part of this test process. Similarly, any other operating systems, browser or platforms such as mobile tablets, must be evaluated using other testing procedures.

Issues that are Not Covered in this Test Process

Problems may be found during testing that impact accessibility, but are simply coding errors. Included here would be things like links that lead to the wrong target website. A tester may notify a developer of these issues as a comment on a report, but they will not typically result in a compliance failure.

The Rationale for Each Test

Each step of the test process includes only the information that will need to be referenced frequently, namely the directions on how to test, and how to interpret the test results. To simplify the organization of this document, the rationale for each individual test has been separated out, and is presented below.

1. Interactive Interface Elements

Interactive elements of interfaces include any elements that a user is expected to use, modify, or edit. Examples include navigation controls (links, buttons etc.), and editable content (selectable text, data input etc.).

It must also be possible for users to determine what those controls are, and how to use them. This requires that the visual label / instructions are programmatically associated with controls; otherwise, non-visual users will not be able to tell which label relates to which control / form-element.

1.1. Keyboard Access

This keyboard-only is a test of whether it is possible to control the interface without the visual and/or physical capabilities necessary to use a mouse.

Wherever users are expected to interact with components, it must be possible for users to get to those components or perform those functions using only the keyboard, because (i) using a mouse is not possible when the user has no sight, and (ii) using a mouse is not possible when the user does not have the physical capability / dexterity to effectively control a pointing device.

Keyboard access is defined as use with physical keyboard that is attached to the computer, either separately (desktop PC) or integrated (laptop PC). Alternatives for keyboard access, such as using an on-screen keyboard, or using the Mouse-Keys feature in Windows, are specifically excluded from this test.

Ideally, interfaces use standard keyboard commands (TAB, Space Bar, Enter, Escape, etc.), making their use easy and efficient. On occasions, an interface may be designed to expand on the basic set of standard keyboard commands; and/or remap standard keys. In both of these cases, users must learn the non-standard keys. In order to be aware of non-standard key commands, users must be notified of their existence and correct use through the interface, application help, and/or documentation.

When controlling the interface with keyboard only, if there is no visual differentiation between the current focused item and the rest of the interface / content, then it is not possible to tell where in the interface you are. Therefore, a visual indication of focus is necessary.

A logical order and groupings of interface components is normally a given in the design of software applications and Web content. Groupings and order are usually visually apparent. Logical arrangements are used to aid visual appeal and improve usability. However, when the focus/TAB order does not follow the logical order, users can become confused, make errors, and may not understand the contextual meaning of components. This is especially true for people who have no vision, or who have low vision, and are relying on AT.

Some components on web content and software screens are intentionally hidden to reduce visual clutter. Other components only appear as part of a procedure such as an error notification. It is possible to reveal content with interface components in an inaccessible manner by requiring user vision and/or requiring the use of a mouse. Keyboard users need to be able to get to the information and controls that are revealed, and users without vision, or with low vision, need to know that new content has appeared.

1.2. Labels for Assistive Technologies

Assistive technology utilizes accessibility properties of elements and provides them to users through various modes to provide access to the application. Accessibility properties are provided through Microsoft Active Accessibility (MSAA), Microsoft User Interface (UI) Automation, Java Accessibility Application Programming Interface (API), HTML code, ARIA properties or other protocol. Without these labels and accessibility properties, the assistive technology may not provide correct information to its user.

1.2.1. SW: All Interactive elements

Interactive software elements include navigation controls (menus, buttons etc.), and editable form fields (selectable text, data input etc.).

To correctly and accurately complete a form, it is necessary to enter information in the correct places and to follow instructions, directions, and cues. A given form component may be the subject of instructions that are not positioned next to the component (e.g., at the top of a form, the instruction is "If you are the home owner, complete parts a, b, and f"). In such cases, form designers will use visual layout and flow to direct the user. However, users without vision, or with low vision, may not have access to the visual cues, and hence will be unable to easily find the related instructions for the current form component. For this reason, it is necessary to programmatically associate all relevant instructions, directions and cues with their respective components/controls.

It is required to use meaningful and unique names for links and user controls, to aid navigation and use by AT.

Certain components in an interface can change their state. States include such things as closed/open, ascending-order/descending-order, collapsed/expanded. Dynamic values can also be shown on components (e.g., "34 characters remaining", "Alert Priority 5"). The current state and function of interface components is usually visually apparent. However, these characteristics of the component must be provided and discoverable by assistive technology for users without vision or with low vision (including without color perception).

1.2.2. Web: Forms

Web forms include controls (checkboxes, radio buttons etc.), and editable content (selectable text, data input etc.).

In order to correctly and accurately complete a form, it is necessary to follow instructions, directions and cues, as well as enter information in the correct places. A given form component may be the subject of instructions that are not positioned next to the component (e.g., at the top of a form, the instruction is "If you are the home owner, complete parts a, b, and f"). In such cases, form designers will use visual layout and flow to direct the user. However, users without vision, or with low vision, may not have access to the visual cues, and hence will be unable to easily find the related instructions for the current form component. For this reason, it is necessary to programmatically associate all relevant instructions, directions and cues with their respective components/controls for AT users.

1.2.3. Web: Links and User Controls

To aid navigation with screen reading AT software, users can call up a list of links on a web page or software screen. Users can read content and decide which of the links in the content they wish to follow (i.e., they do not have to navigate back to the link itself).

In order to provide links to end users, there are a number of common methods in practice that render a list of links unhelpful. Say each item for sale has a 'click here' link next to it, and the user calls up the list of links. The list will have multiple 'click here' links that are not distinguishable. Another common problem occurs when the links only contain URLs, and the purpose of each link may not be apparent.

It is therefore required to use meaningful and unique names for links and user controls, to aid navigation and assistive technology for users without vision or with low vision.

2. Non-Text interface elements

Non-text elements include images, text rendered as an image, graphs, charts, audio files, animations and video files.

Screen reader software cannot interpret images, animation, or video. Screen readers will, however read text that has been associated with interface elements. The interpretation (meaning) of an interface element must therefore be conveyed textually in the interface programming for .assistive technology for users without vision or with low vision.

2.1. SW: Images

Software images include controls (links, buttons etc.) that are rendered as images, static images, charts, diagrams, text rendered as an image, etc. Essentially, all non-interactive software elements that provide meaningful information must be conveyed correctly to assistive technology through their accessibility properties (Name, Role, State) for assistive technology for users without vision or with low vision.

2.2. Web: Images

Web images include controls (links, buttons etc.) that are rendered as images, static images, charts, diagrams, text rendered as an image, etc. Equivalent descriptions must be provided for all images for assistive technology for users without vision or with low vision.

2.3. Video-only and Animation

Software animation includes sequences of overlaid images, dynamic changes of state such as a moving speed dial, a chart illustrating dynamic flow changes from one state to another, etc. Video-only files include animations, screen video captures etc. The visual information provided through these file types must be provided through alternative means for assistive technology for users without vision or with low vision.

2.4. Web: Audio-only

Audio-only files include speeches, sound-bites, ambient (background) sounds, etc. Equivalent text descriptions must be provided for users with no hearing or hard of hearing.

2.5. Web: Image Maps

An image map is a single image that has designated regions or "hotspots" that contain links. Server-side image maps cannot be made accessible for keyboard users or assistive technology users.

3. Color and Contrast

The use of color to convey meaningful information must be provided through alternative means of users who cannot distinguish colors. Insufficient color contrast may make it difficult for some users to see and use the content.

3.1. Color Dependence

Color dependence is using color as the sole means to convey information. For example, a single unlabeled indicator that is green for 'on', orange for 'standby', and red for 'off' is color dependent.

When color is the only means to convey information, people who are color blind, and people who cannot see, do not have access to the same information that others have. The status or function that is being conveyed by color also needs to be available in a textual format that can be viewed by all, and can be read by screen reader software.

This requirement does not mean that color cannot be used; it means that color cannot be the *only* means of conveying the information.

3.2. Color Contrast

The visual difference between the background behind text, and the text itself, may be perceivable by a given designer. However, beyond color choice which is under control of the designer, many factors beyond the designer's control affect peoples' ability to discern between colors/shades, including age (contrast sensitivity reduces with age),screen brightness, ambient light, color blindness and some types of low vision. The use of color/shade choices that do not contrast well with each other may be deliberate (i.e., artistic preference), or they may be the result of programmatic features (e.g., a button's text is black on white, but the text turns yellow in a certain mode, and the background remains white).

In general, the higher the level of contrast used, the more people will be able to see and use the content.

4. Flashing

The term 'flashing' encompasses interface elements that blink, flicker repetitively, or elements that scroll (e.g., marquee text).

An element that flickers or blinks in the visual field can cause adverse reactions/seizures in people who have photosensitive epilepsy. Section 508 does not permit flashing in the frequency range from 2Hz to 55Hz (from twice to 55 times per second). This frequency has been updated in WCAG 2.0 and the baseline tests to not permit flickering above 3 hertz, which is followed by DHS.

Scrolling ('marquee') text should be avoided where possible, because the scrolling effect actually causes a form of flickering that can be in the prohibited frequency range. Even though this may be imperceptible for many viewers, it can have the same flickering effect for some.

5. Page Titles

Page titles appear in the title bar of the browser or software window (and in the tabs where multiple tabs in a single window are used). If there is no programmatically defined page title, visually capable users can assimilate the content quickly to know where they are. However, non-visual users will have to navigate through the content to know what page they are on, which can take an undue amount of time.

Screen reader technologies will announce the programmatically defined page title when the user is browsing between tabs and between windows.

6. Multimedia

Synchronized media is a presentation consisting of time-synchronized video and audio. Synchronized media includes public information films, Web casts, press conferences, and online training presentations.

Some users will not be able to hear the content well or hear it at all. Therefore, there needs to be another mode to provide the audio information. This usually means captions (text showing what is being said, and other relevant sounds). Captions need to be available, but do not necessarily need to be turned on by default. For example, users who need captions can switch them on with a control (usually a 'CC' button for Closed Captions). If there is no means of switching modes, then the default mode must be accessible (i.e., Open Captions).

Because captions must be time-synchronized, separate transcripts will not meet this requirement on their own.

Some users will not be able to see the content well or see it at all. Therefore, there needs to be another mode to provide descriptions of the visual information. In synchronized media, this usually means additional narration inserted during breaks in the dialog, describing visual events and cues.

Audio descriptions need to be available, but are not required to be turned on by default. For example, users who need descriptions can switch them on with a control. If there is no means of switching modes, then the audio descriptions must be enabled by default.

The Alternative presentation of information must allow understanding of the relevant information. For example, descriptions might include the looks on people's faces, people handing items to each other, or who has entered the room.

7. Time Outs

Messages and/or instructions to the user requesting their response within a given time are typically associated with sites that require a secure login. This includes both server time-outs and client-side security time-outs.

People who use AT such as screen reader software or voice input software may require more time than other users to assimilate the information and execute the controls on a Web page or software application. Because AT users may need more time, applications that have a time out must provide (a) prior notification/warning that a time out is about to occur, and (b) a means for the user to request more time.

8. SW: Built-in Accessibility Features

It is possible to write software that controls various aspects of the OS. The control commands may inadvertently cause an OS accessibility feature to deactivate. For example, Sticky Keys is a feature that enables users to use one finger or pointer to use the control key, the alt key, and the shift key sequentially (rather than the simultaneously, as they are ordinarily used). If a developer wanted to reset the keyboard state because they wanted to turn off the CAPS LOCK indicator, they must take care not to reset accessibility features at the same time.

It is also possible to override OS accessibility appearance features. For example, in High Contrast mode, the color settings of standard windows components are modified throughout the OS. If a developer wanted to make a special green and brown camouflaged theme for his or her application, they would specify the exact colors that would be used in the menus and other window control components. By specifying the exact colors, rather than adopting system colors, they override the ability of the user to employ the high contrast settings that they need to access the application. In this example, an aesthetic preference results in non-compliance with the requirement not to interfere with the user's appearance settings.

The accessibility features of Windows 7 (the platform for which the baseline tests are written) contain the following user-configurable accessibility features that should not be disabled or disrupted by the software application:

- All settings in the Ease of Access control panel
- System color settings, including high contrast modes
- System text size settings

This requirement also applies to software that is embedded in a page displayed in a Web browser.

9. Web: Text Properties

Web content, including text of the page, can be difficult for screen reader users to comprehend if certain attributes and structure are not provided.

9.1. Web: Language

When a site is in one language but has certain pages that are in a different language, or sections within a page that use a different language, it is necessary to programmatically identify both the default language and the change of language. Screen reader technologies can switch their language pronunciation, but only if there is code to identify the proper language. If language changes are not identified for a screen reader user, the speech will sound awkward at best or unintelligible at worst.

9.2. Web: Section Headings

Headings are used to visually and semantically break up content to make it easier to read, easier to find and understand relevant information, and so on. Headings can be visually marked using text formatting such as bold, underline, or combinations (e.g., bold, underlined, and large font means a major heading).

Screen reader technologies cannot automatically infer meaning from formatting changes. A given piece of text may be in italics because it is emphasizing a point, or because it is a heading. Because there is no way to infer meaning, headings can use visual formatting, but they must also be programmatically identified for identification with AT.

- The requirement should not be construed to require headings in place of headers in data tables.
- This requirement does not mean that headings need to be added; it means that where headings are identifiable through visual formatting, they must be programmatically identified.

Any visual representations of heading level (e.g., major section, section, subsection) must be matched by the programmatic heading level (i.e., major section = level 1, section = level 2, sub-section = level 3). Matching the programmatic level with the visual level is essential for proper comprehension of the content for non-visual users.

10. Web: Data Tables

For users with vision, the process of determining what headers go with a data cell is usually straightforward, especially when formatting such as bold letters and shading is applied to the headers. For users of screen reading software, however, things like 'bold' and 'shaded' have no useful meaning, so using styles and formatting to identify headers does not work. Instead, row and column headers must have programmatic markup to enable them to be identified by the screen reading software.

On all data tables, column and row headers must be identified.

Data tables are those tables where the information in a cell requires a row or column header to adequately describe the cell's contents. If a table is used for placement of components on the page for visual aesthetics, then it is a layout table. This test applies to data tables only.

Complex data tables are defined as those that have two or more levels of headers, and/or include split or merged cells. On complex tables, markup is needed to define which headers are associated with data cells, so that screen reader users can determine where they are for any given cell or set of cells.

11. Web: Style Sheet Dependence

Style sheets are a means to provide visual formatting information to complement a Web page's content.

The original intention behind style sheets was to separate presentation from content. The text, images, links etc. comprise the 'content', and things such as font choice, background color, link underlining etc. comprise the presentation 'style'.

A Web page should in theory always be readable and functional without the developer's style sheet, since content is separate from presentation. However, it is possible for developers to inadvertently deliver content through style. For example, a background image can be applied with a style sheet, but if that background image also contains important information, such as an organization's name, logo and contact details, then content is no longer separate from presentation.

Because of their particular visual needs, some people with visual impairments create their own style sheets (font color, background color, etc.) to replace the provided style sheet. When content is not properly separated from presentation, it becomes difficult or impossible to read the information on the screen. Therefore, pages must be tested with style sheet information removed, to ensure that all content is still being delivered to the user.

12. Web: Frames

Frames are a means of separating out sections of a Web page into different navigable regions.

To mouse users, the separation of a Web page into sections means that they can scroll the information in one frame without affecting another frame. Keyboard only users who are able to see can navigate between frames (F6 key is the browser default for this function).

Non-visual users can also use the keyboard to navigate between frames, but if there is no programmatic name for the frames, the user has to read the content of each frame in an attempt to discern where the information they need might be. This can take a long time, and can lead non-visual users to make errors. For this reason, it is necessary for each frame to include a descriptive name. The name should make sense when spoken in isolation as the user navigates between frames.

13. Web: Repetitive Content and Links

Groups of navigation links are usually provided along the top and/or left of multiple pages to provide quick navigation to other areas of a Web site. In addition, some groups of pages may repeat blocks of content (other than navigational controls).

For users who can see and use a mouse, skipping over navigation links and other blocks of content is simply a mouse movement followed by a click. However, for users who cannot use a mouse, repetitive links can be a serious impediment to productivity. If a site has forty repetitive links, a keyboard user must complete forty keystrokes just to get to the information they need to use on every page.

To enable equitable use by keyboard only users, there must be a method to skip past repetitive content. This can be provided by adding internal links to bypass repetitive content. Similarly, for screen reader users, if they must read content that is repeated on each page and cannot skip past it, their experience on the page can be very frustrating.

14. Web: Required Plug-ins

It may be necessary or desirable to deliver content that must be displayed in a separate application, or via a browser plug-in, such as PDF and other electronic document files, and synchronized media.

For public facing Web pages, there must be a link provided either directly (i.e., next to the content) or indirectly (i.e., a page providing links to all plug-ins used on a site).

The plug-in must meet all applicable software requirements.

15. Web: Alternative (Accessible) Version

An ' Alternate Page' is an accessible version containing the same information as the primary page. Alternate pages will usually contain text in place of the inaccessible content from the primary page. For example, a complex organizational chart may be written in prose. The text must be equivalent, and it must be kept up-to-date.

An ' Alternate Page' should only be provided for accessibility when the primary page cannot be made accessible. The accessible version must contain the same information as the primary page.

The information should be 'equivalent', but by definition, this is not going to be 'exactly the same'. The main points, themes, concepts etc. that the authors are trying to get across in the primary content should also come across in the alternate page. For example, if a complex chart on the primary page shows a year with a small increases in earnings in Q2 and a large decrease in Q2, and the text discusses why these trends seem to be occurring, the Alternate page should convey the trends, and the high and low data points of interest. An alternate page that just gave all the data points in linear form, with no highlighting of the trends under consideration, would not be considered equivalent.

Alternative versions of pages for accessibility are only permitted when the primary page cannot be made compliant. Common examples where alternative versions are usually permitted include maps and directions, and very complex diagrams and charts.

Section 2:
Test Environment

The approved test environment is described in this section. Only the specified operating systems and version(s) of testing tools are supported by this test process. As browsers are frequently updated, it may not be possible for the documentation to be up to date with the latest browser versions. If any update creates critical issues for test procedures or results, that specific version may be not be supported or test instruction edits may be necessary. Critical issues and modifications will be published as quickly as possible. Please refer to the **Trusted Tester Test Environment Installation and Configuration Guide** at https://www.dhs.gov/dhs-section-508-compliance-testing-tools for additional details. The Guide contains the most up to date test environment information and supersedes any conflicts with this section of the document.

Windows OS

This test process contains instructions for testing applications running in the following operating systems:

- Windows 7,
- Windows 8.1 (desktop mode), and
- Windows 10 (desktop mode).

Browsers

This test process supports testing of Web applications in the following browsers in supported operating systems:

- Microsoft Internet Explorer 11,
- Google Chrome, and
- Mozilla Firefox.

Use of newer versions of these browsers is acceptable unless otherwise specified in the **Trusted Tester Test Environment Installation and Configuration Guide** at https://www.dhs.gov/dhs-section-508-compliance-testing-tools.

Most Accessible Browser for Testing: IE11

There are notable differences in test results depending on the test environment. Specifically, certain browsers provide varying levels of support for certain code techniques. Browser differences may cause test results to vary for an application. Test results for the test environment should be reported as instructed (NC| C |NA) in each baseline test. It may be necessary to test an application in multiple test environments in performing root cause analysis of issues and/or remediation.

While this test process supports multiple browsers, IE11 is the most accessible test environment for Web applications that contain Flash or embedded Java elements. Due to Chrome and Firefox's diminishing levels of support for Flash and Java, these browsers may not fully reveal the coded accessibility properties for these content types. Testing may still be performed in Chrome and Firefox to determine results in that specific browser. However, to determine the compliance of the coded content, Web applications that contain Flash or Java components should be tested entirely in IE11. These caveats will be repeated in this document where relevant.

If it is unknown, at the start of testing, whether a Web application contains Flash or Java, the general recommendation is to test in IE11. Testing of applications that do not contain Flash or Java components can be performed in any of the approved test environments.

Configure Browsers for Testing

Some browsers require set up for accurate test results. Please refer to the **Trusted Tester Test Environment Installation and Configuration Guide** at https://www.dhs.gov/dhs-section-508-compliance-testing-tools.

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 use. The tools assist the tester with verification of HTML code and accessibility application programming interfaces (APIs). 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.

The following are the testing tools used in this test process. For version, installation, and setup information, please refer to the **Trusted Tester Test Environment Installation and Configuration Guide** at https://www.dhs.gov/dhs-section-508-compliance-testing-tools.

Inspect

Official name	Inspect
Also Known As	Object Inspector 32, Inspect Objects
Purpose for 508 testing	Reveals the Name, Role, State and Value of software elements.
Developed by / Owned by	Microsoft Corporation

Java Ferret

Official name	Java Access Bridge (for the main software) Java Ferret (for the component within the main suite)
Also Known As	Java Access Bridge
Purpose for 508 testing	Reveals the Name, Role, State and Value of Java software elements
Developed by / Owned by	Oracle Corporation

Web Accessibility Toolbar (WAT)

Official name	Web Accessibility Toolbar for IE
Also Known As	Web Accessibility Toolbar - IE Plug-in WAT Toolbar WAT-C Toolbar
Purpose for 508 testing	Adds a toolbar to Internet Explorer to aid manual inspection of accessibility related elements on web pages.
Developed by / Owned by	Web Accessibility Tools Consortium

ARIA Markup Favelet

Official name	ARIA Markup
Also Known As	Jim Thatcher's Favelets
Purpose for 508 testing	Adds a JavaScript function that highlights all ARIA code on the page. Supplemental favelet for use with WAT; already included with WAF (see below).
Developed by / Owned by	Jim Thatcher

Frames Favelet

Official name	Frames
Also Known As	Jim Thatcher's Favelets
Purpose for 508 testing	Adds a JavaScript function that marks up frames and iframes on the page with name/title. Supplemental favelet for use with WAT; already included with WAF (see below)..
Developed by / Owned by	Jim Thatcher

Web Accessibility Favelets (WAF)

Official name	Web Accessibility Favelets (WAF)
Also Known As	WAF
Purpose for 508 testing	An alternative to WAT that can be installed in IE11, Firefox and Chrome. WAF includes the ARIA, Frames, and Skip Link favelets from Jim Thatcher.
Developed by / Owned by	N/A (Open Source)

Notes on WAF

WAF is a set of favelets that can be easily installed on various browsers. The baseline tests were originally developed using WAT, a browser plugin that is only compatible with IE. To include other browsers in the approved test environment and minimize changes to test instructions, WAF, a suite of JavaScript favelets, was developed to replicate WAT operation and functionality as closely as possible. Several Jim Thatcher accessibility favelets have been included in WAF to optimize functionality. Alternative instructions are provided for WAF where there is a significant difference from WAT in terms of testing or interpreting results.

The functions of the WAF favelets were named according to their corresponding functions in WAT, minus the menu hierarchy. For example, where the WAT instruction is

> Use the **WAT (Doc Info – Show Titles, Images - Show Images)** to examine …

If using WAF, interpret these instructions as

> Use the **WAF (Show Titles, Show Images)** to examine …

Once installed, the tester can organize the favelets by any preference (e.g. in alphabetical order, in subfolders, in testing order, etc.).

WAF nuances:

1. Disabling CSS: Whereas **WAT (IE - Toggle CSS)** disables or enables CSS for all pages and tabs in IE11 until this function is re-selected, **WAF (Toggle CSS)** only disables CSS for one page. Additionally, refresh the page to reactivate the CSS for the page instead of reselecting the **WAF (Toggle CSS)** favelet.
2. Unlike WAT, WAF does not include the Colour Contrast Analyzer. The standalone version of the Colour Contrast Analyzer should be installed with WAF.

These caveats are repeated in this document where relevant.

Colour Contrast Analyzer

Official name	Colour Contrast Analyzer
Also Known As	CCA
Purpose for 508 testing	A standalone version of the Colour Contrast Analyzer for use with WAF; (included in WAT)
Developed by / Owned by	JUn, Steve Faulkner

Selecting the Correct Testing Tools

For the purposes of this test process, we classify applications as either Web applications or stand-alone software (or just "software") applications. It is important to identify whether the interface element you are testing is software or Web so that you know which tool to use and what test outcomes are expected. If you try a tool and it does not display the test results properly, try another tool. If Inspect does not display the information, try Java Ferret, and vice-versa.

For reference,

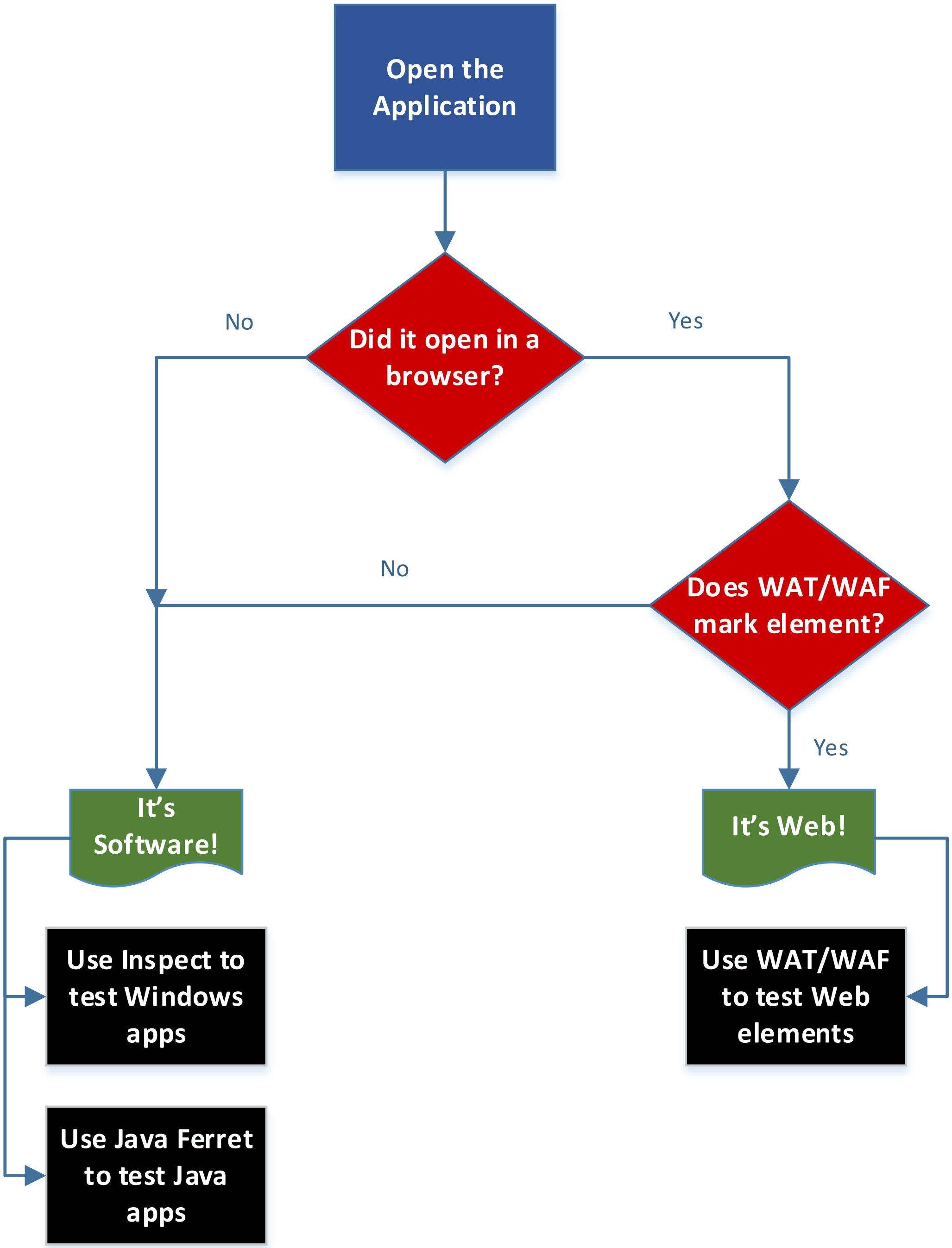
- Software interface elements are those delivered to the user via native operating system-based processes.
- Web interface elements are delivered to the user via a Web browser.
- Browser plug-ins are examples of Software interface elements embedded in Web pages.
- Browser-based help pages are examples of Web page elements embedded in Software.

Some standards (such as keyboard access) are listed in more than one test. Through the testing process, ***a test result can change from Compliant to Not Compliant, but do not change from Not Compliant to Compliant*** or your results will become inaccurate.

Which Testing Tool Should I use?

** Use of "WAT" in the instructions below encompasses any of the Web testing tools (WAT, WAF, favelets). For a visual flowchart, refer to the illustration on the next page.*

- Open the Application.
- If it does not open in a browser, (e.g. Microsoft Internet Explorer 11), treat it as "software." Use the software tools: Inspect (Object Inspector) for Windows applications and Java Ferret for Java applications. Always check the other tool if one tool does not display accessibility information.
- If it opens in a browser and the Web Accessibility Toolbar (WAT) or the favelets successfully marks up the element, apply the Web standards to that element.
    - If the WAT function does not provide its markup on an element, it may be an embedded software element. If it is, use the software tools and apply the software standards.
    - Web applications with embedded Flash or Java elements must be tested using IE11.


Open the Application
Did it open in a browser?
No
Yes
Does WAT/WAF mark element?
No
Yes
It's Software!
It's Web!
Use Inspect to test Windows apps
Use Java Ferret to test Java apps
Use WAT/WAF to test Web elements

Note: Web Applications with Flash or Java elements must be tested using IE11.

Testing Workstation setup

While application testing can be performed on a laptop, the recommended testing workstation includes the following computer peripherals:

1. Dual monitors (to test on one screen and enter test results on the other)
2. Full-size Keyboard
3. Mouse
4. External speakers (enable sounds and volume adjustment)

Section 3:
Section 508 Compliance Tests for Applications

Structure of this Section

For consistency, each of the test processes has the following information:

- the directions on **how to test**, and
- how to interpret the **test results**, and
- the **applicable standards**.

The following are how parts of this process are referenced:

Test Process 1 – Interactive Interface Elements (main category)

Test 1.1 Keyboard Access (sub-category)

Step 1a (instructions)

Test ID 1.1.A.(failure condition)

Note that some test processes only include one Test instruction and are only referenced as the test process.

To simplify the organization of this document:

- the **rationale** for each individual test has been separated out (see page 4).

1. Interactive Interface Elements

Interactive elements of interfaces include navigation controls (links, buttons etc.), and editable content (text input fields, select options, etc.).

1.1. Keyboard Access

Keyboard access and control must be available for all interactive interface components and frames that can be accessed or controlled with a mouse. Where non-standard keyboard commands are employed, users must be notified of alternate keyboard commands through the interface, application help, and/or documentation.

For a user to know where the keyboard focus is on the screen, there must be a visible indication of the currently focused component. If hidden content is revealed on the screen, the keyboard focus must move to the new content. When the sequence of interface components has meaning or requires an order of operation, the focus/TAB order must logically flow with the application/content.

Keyboard access is defined as use with physical keyboard that is attached to the computer, either separately (desktop PC) or integrated (laptop PC).

- ***On-screen keyboards, and the Mouse-Keys feature in Windows cannot be used for this test***

1: Keyboard tasks - How to Test

1a Find all visible and hidden interactive interface components that are accessible by mouse (hover and/or click). Note: Web applications with interactive Flash and embedded Java content must be tested in IE11 because they create keyboard traps in Firefox and Chrome.

1b Use the Windows keyboard shortcuts (Tab, [Shift+Tab], Space bar, ALT, arrow keys, Enter, etc.) to navigate through the entire screen to each interactive interface component, reveal hidden content, and activate all interactive components.

Use only the physical keyboard to access all interactive elements and functions of the application. Only elements that can be accessed or controlled with a mouse must be keyboard accessible.

Note if the keyboard gets "trapped" and appears to loop or get stuck within an area of the application as a failure. Use the mouse only in this situation to bypass the trap and continue testing.

1c Investigate any instances where the standard Windows keyboard methods do not provide access to interactive elements. **Check if alternative keyboard access instructions are provided** (in the application's Help section or within the page). Some applications and web content will display keyboard functions when the ALT key is pressed. Instructions may include extending standard keyboard command operations (e.g., getting out of a keyboard "trap"), and/or alternate keys. Verify that the alternative keyboard commands work.

Some applications may offer multiple ways to activate a function. Only one method of keyboard access is required. Special access keys (such as [Alt+O] or [Control+P]) and redundant menu options are not required for compliance if Windows shortcuts work.

1d **Complete form fields**. Enter text, select (arrow and tab) different options from drop down lists (ALT+down arrow), select and unselect (spacebar and arrow keys) checkboxes and radio buttons. Try to select the second or third option from a drop down list by keyboard only. Where multiple selections in a drop down list are possible, select the second, third and fifth options or other combinations. Reference help for keyboard access instructions or assistance. Hidden and disabled form fields are not interactive and do not require keyboard access.

1e **Use WAT (Doc Info-Show Titles)** to reveal information that will be revealed by mouse over through the TITLE attribute. Inspect the information provided through the TITLE attribute on all interactive components (including interactive images).

Check if the TITLE information can be revealed by placing the keyboard focus on the element. If the information in the TITLE attribute cannot be revealed by keyboard, it must be conveyed through screen text or visual context.

2: Focus tasks – How to Test

2a **Follow the application's visual focus that moves with keyboard navigation.** Visual focus is an indication of the keyboard focus location. Only keyboard accessible elements need to be analyzed for visual focus. This is typically a dotted rectangle, highlight, or a vertical bar in a text field. Do not enhance the visual focus with any tools for this test.

It is not a requirement that only interactive elements receive focus. For example, form instructions in text may receive focus by design. This is not a failure.

2b **In forms,** the focus may automatically shift to the next form field once valid text is entered (e.g. after three numbers are entered for an area code, the focus immediately shifts to the field for the next three numbers of a phone number.) The focus must be visible after shifting.

2c When hidden content is revealed, check that the visual focus was moved to the new content. (Information revealed by TITLE (step 1e) should not be part of this test.)

Move the focus to the control that reveals hidden content, activate the control with the keyboard, and then determine whether focus is in the revealed content. It may be necessary to TAB once to find the focus.

If a modal dialog box (requires user interaction to return to the application) is revealed, the visual focus must remain within the modal dialog box until the box is closed. If a dialog box behaves like a modal dialog box when using the mouse, test it as a modal dialog box.

(If focus does not move to the revealed content, an accurate description of the content change must be provided. This description may be revealed in many ways: on screen text, tooltips, a description on the element that reveals the hidden content. See Sections for images (2.1, 2.2), labels for assistive technologies (1.2) to test for descriptions.

2d Note any instances where the focus order (also called tab order) is illogical. If the visual focus does not move through the elements in a logical order, determine if this can cause confusion or errors for a user. Consult with application developer to confirm if tab order is intentional, if necessary. Evaluate the existing focus order only; do not consider elements that do not receive keyboard focus.

2e Note any instances of a loss of visual focus (can't see or find the focus on the screen) while Tabbing through the page or if the visual focus appears on the wrong element as a failure. Loss of focus should not occur while manually shifting focus through the page (using the TAB or arrow keys). On a page that has frames, use the <F6> key to shift focus to each of the frames. The focus must be visible when it is on the frame.

While manually shifting focus through the page, the target anchor of a skip link must not cause a loss of focus.

When a function that moves the focus is executed (such as from the skip link itself, or when hidden content is revealed), it may be necessary to press TAB so focus becomes visible again. This is not considered a failure.

Note: Refresh (F5) the page to remove WAT markup.

1.1. Results

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. An interactive element or function cannot be accessed or activated by keyboard [21(a), B1]	Test ID always applies. **[DNA] is not an acceptable result**.	**[NC]** if an interactive element cannot be accessed by keyboard.	**[C]** if all interactive elements can be accessed by keyboard.
B. A keyboard "trap" is found [21(a), B1]	Test ID always applies. **[DNA] is not an acceptable result**.	**[NC]** if a keyboard "trap" is found. **(Note: this will cause 1.1.I to be NC).**	**[C]** if a keyboard "trap" is not found.
C. Non-standard or alternative keyboard commands are required for access but are not documented [21(a), B1]	**[DNA]** if all interactive elements can be accessed with the standard keyboard commands. (e.g., tab, arrow keys, enter, space, etc.)	**[NC]** if non-standard keyboard commands are needed to access an interactive element and the commands are not documented.	**[C]** if non-standard keyboard commands are needed and they are documented.
D. Information revealed by mouse over (TITLE) is not available to keyboard-only users (i.e., there is no equivalent screen text or visual context) [21(a), B1]	**[DNA]** if mouseovers (TITLE) are not used.	**[NC]** if information is in the TITLE but is not represented through text or visual context.	**[C]** if TITLE provides information and equivalent information is found through text or visual context.
E. At any time, there is no visual indication of the current focus (loss of focus) [21(c), B2]	Test ID always applies. **[DNA] is not an acceptable result**.	**[NC]** if a loss of visual focus occurs.	**[C]** if the visual focus can be determined at all times.
F. The visual focus appears on the wrong element [21(c), B2]	Test ID always applies. **[DNA] is not an acceptable result**.	**[NC]** if focus appears on the wrong element.	**[C]** if focus always appears on the element it is programmatically on.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
G. The visual focus does not remain within a modal dialog box until closed [21(c), B4]	[DNA] if there are no modal dialog boxes.	[NC] if focus leaves the modal dialog box while the box is open.	[C] if focus remains in the modal dialog box until the box is closed.
H. The visual focus does not move to revealed content and no description of the content change is provided [31(a), 31(b), B4]	[DNA] if there is no revealed content.	[NC] if the focus does not move to the revealed content AND a description of the content change is not provided.	[C] if focus moves to revealed content **OR** a description of the content change is provided.
I. The tab order is not logical [31(a), 31(b), B3]	Test ID always applies. **[DNA] is not an acceptable result**.	[NC] if the tab order is not logical. **(Note: this will be NC if 1.1.B is NC.)**	[C] if the tab order is logical.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

21(a) Keyboard Accessibility

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

21(c) Visual Focus

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

31(a) Use without Vision

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

31(b) Use with Low vision

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

Applicable Baseline Requirements

B1. Keyboard Navigation

Keyboard access and control must be available for all interactive interface components and frames that can be accessed or controlled with a mouse. Where non-standard keyboard commands are employed, notify users of the existence and use of Alternate keyboard commands through the interface, application help, and/or documentation.

B2. Focus (visible)

There must be a visible indication of the currently focused interactive component.

B3. Focus (order)

When the sequence of interface components has meaning or requires an order of operation, the focus/TAB order must logically flow with the application/content.

B4. Focus (revealing hidden content)

Components that reveal hidden content (dialog boxes, light boxes, pop-ups, content accordions, drop-down menus, etc.) must either (i) shift focus to the content they reveal, or (ii) the component must describe that a change to the content will occur if selected.

1.2. Labels for Assistive Technologies

Descriptive information must be provided for all interface elements to enable assistive technologies to pass useful information on to its user. The descriptive label must include identification, status and purpose of an element.

1.2.1. SW: All interactive elements

All software elements must have useful and complete descriptions.

How to Test	
1	**Find all interactive interface elements such as user controls (buttons and menus) and form field elements. For forms fields, also find all associated instructions and cues, including whether they are required fields**
2	**Start Inspect or Java Ferret.** Use the keyboard to navigate to each element. If an element is not keyboard accessible, use the mouse to inspect it.
3a	**Inspect the *Name, Role, Value*,* and *State*** for each element with Inspect/Java Ferret. The testing tool data will update with each focus change.
3b	**The *Name* of each element should match the visual information.** If there are multiple functions with the same visual label, check that the Inspect/Java Ferret Name is distinct. For example, if there are multiple "Submit" buttons on a screen, the Name for each must be unique – "Submit time of departure", "Submit time of return". Unique functions must have unique Name properties and describe the action it will perform. In the Inspect tool (UI Automation), check the "Name" or "LegacyIAccessible.Name" property. Inspect form fields to check if all instructions and cues (textual and graphical) are provided. The Name of each component should match the visual label. Any required form fields must be indicated in either Name or State.
3c	**The *Role* of an element must accurately reflect its function.** For example, a menu item must not have a Role of editable text. In the Inspect tool (UI Automation) tool, check the "LegacyIAccessible.Role" property.
3d	**The *Value** property must be correct for text input form fields in Inspect.** To test, it may be necessary to type into the form element, [Tab] out of the element, then [Shift+Tab] back to the field. The Value field will be updated with the focus change and should contain the same text as the input field. In the Inspect tool (UI Automation), check the “LegacyIAccessible.Value" property. In Java Ferret, check the "Value" or "Sentence" property.

3e **The State of interface elements with focus must be "focused, focusable".** Grayed out interface elements must be correctly identified as "unavailable" or similar. The state of a selected checkbox/radio button must be correctly listed as checked or selected. (Windows applications will not reveal "unchecked" or "unselected". Java applications will.) Again, it may be necessary to [Tab] out and [Shift+Tab] back to the field to update the State in Inspect/Java Ferret.

In the Inspect tool (UI Automation), check the “LegacyIAccessible.State" property.

4 **Elements that have variable statuses must reveal their current status.** This information can be provided through the Name or State property. For example, if a text field has a character limit, the Name property may be "Describe the Damage to Home (34 Characters Remaining)" or the State of that form field may be "focused, 34 characters remaining". This information may also be in the "description" property. It may be necessary to force the status to change to check that this is working properly. (Do not include radio buttons and check boxes in this test.)

Other examples of statuses include sort order, opened or closed menus, expanded or collapsed tree views.

Complete status information is required. If a data table offers multiple sort options (for example, sortable by date, last name, and city), the data table's current status must include which column is the primary sort option and how that column is sorted. It is not required that a single component provide the complete status for a component (e.g., an asterisk can indicate the column that is the primary sort and a down arrow can indicate that it is sorted alphabetically A to Z).

1.2.1. Results

Note: If the application is not software, mark these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. A user control does not have a descriptive and unique Name property [21(d), B16]	Test ID only applies to software applications. [DNA] if there are no user controls.	[NC] if a user control does not have a descriptive and unique NAME property.	[C] if all user controls have a descriptive and unique NAME property.
B. A user control has an incorrect Role and/or State [21(d), B16]	Test ID only applies to software applications. Test ID always applies when 1.2.1.A applies. [DNA] if there are no user controls.	[NC] if the Role or State of a user control is incorrect.	[C] if the Role and State of all user controls are correct.
C. An interactive interface element that has multiple statuses does not indicate its current status [21(d), B6]	Test ID only applies to software applications. [DNA] if no interface elements have multiple statuses.	[NC] if an interface element with multiple statuses does not indicate its current status in NAME Property.	[C] if all interface elements with multiple statuses indicate their current status in NAME property.
D. An input form (text field, checkbox, radio button, etc.) Name does not match its visual label or does not include complete instructions and cues [21(d), 21(l), B11]	Test ID only applies to software applications. [DNA] if there are no form fields.	[NC] if a form field's NAME property does not match all of its visual labels.	[C] if all form fields have a NAME property that matches all of their visual labels.
E. An input form Role does not accurately reflect its function [21(d), 21(l), B11]	Test ID only applies to software applications. Test ID always applies when 1.2.1.D applies. [DNA] if there are no form fields.	[NC] if a form field's Role property does not accurately reflect its function.	[C] if all form fields have a Role property that accurately reflects their function.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
F. An input form State does not accurately reflect that it is selected or has focus [21(d), 21(l), B11]	Test ID only applies to software applications. Test ID always applies when 1.2.1.D applies. [DNA] if there are no form fields.	[NC] if a form field's State property does not accurately reflect that it is selected or has focus.	[C] if all form fields have a State property that accurately reflects it is selected or has focus.
G. An input text field has an incorrect Value. [21(f), 21(l), B11]	Test ID only applies to software applications. [DNA] if there are no input text form fields.	[NC] if an input text form field has an incorrect Value property.	[C] if all input text form fields have a correct Value property.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

21(d) Name, Role, State

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

21(f) Input Text

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

21(l) Forms

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

Applicable Baseline Requirements

B6. Multi-state components

Components that can change their state must reveal their current state and function to Assistive Technology.

B11. Forms (associated instructions)

Labels, instructions, directions and cues necessary to complete a form must be programmatically associated with their respective input control.

B16. Links and User controls

Links and/or user controls must have meaningful names that describe the unique destination, function, and/or purpose of the control for assistive technology.

1.2.2. Web: Forms

Web forms include controls (checkboxes, radio buttons etc.), and editable content (text input, select options etc.).

How to Test

1 **Find all form fields and identify all of the necessary instructions and cues** for completion of each form field. Some instructions may not be next to the form e.g., at the top of a form, the instruction is "If you are the home owner, complete parts a, b, and f". Similarly, some input fields will have more than one instruction. Each form field must be associated with all instructions and cues. Only one method is necessary (Label and ID, TITLE, or ARIA) to associate all instructions and cues with the form field.

2a **Use WAT (Structure – Fieldset/Labels) AND (DocInfo – Show Titles)** to check HTML forms**.**

2b **Check each input form field for an ID and a corresponding LABEL or a descriptive TITLE.** Fieldset and Legend may be used for grouping and associating two instructions to one input field.

2c Check that **Label and ID are valid HTML**:

- Check that there is **only one** 'Label for' and only one 'ID' assigned to each instruction / input component.
- Every 'ID' on the page must be **unique**.
- The 'Label for' and 'ID' for the pair must be identical (**case sensitive**).
- Determine whether the page is coded in HTML 4.01 or HTML5. Press F12 or view source to inspect the DOCTYPE.
    - <!DOCTYPE HTML> means the page is coded in HTML5.
    - <!DOCTYPE HTML PUBLIC … > means the page is coded in HTML 4.01.
- In HTML 4.01:
    - A valid ID must start with a letter (starting with numbers is prohibited).
    - After the first letter, any combination of letters (a to z, A to Z), digits (0 to 9), hyphens (-), underscores (_), colons (:) and periods (.) is allowed.
- In HTML5:
    - A valid ID
        - must be unique
        - cannot be the empty string (“”)
        - cannot contain spaces.

2d Check any **directions provided with TITLE**. If the directions are only revealed by mouse activity using TITLE, equivalent directions must be available in screen text or visual context for keyboard users. For example, if TITLE is used to instruct the user to enter the date in a specific format, and the TITLE information is not

	revealed for keyboard users, "mm/dd/yyyy" must also be visible next to the form field.
3	Execute the ARIA Favelet to **identify ARIA form attributes on the page**. ARIA (Accessible Rich Internet Applications) defines a method that specifies how to increase the accessibility of web pages, in particular, dynamic content and user interface components. If ARIA is found, check that the ARIA attribute(s) and corresponding ID(s) are correctly identifying the labels for the form element. Note fields which are marked with ‘required=true’.
4	Check that **all directions and cues** are associated with the form field. Typical examples include visual instructions, directions, questions, data format, and required indicators. There are many ways to indicate that a field is required. Usually this is visually indicated by an asterisk (*) or other symbol. This indication must be associated with the input component via one of the above methods (adding a title attribute, including the '*' within the label, or through ARIA ‘required=true’).
5	Complete some form fields incorrectly (wrong format of data entry and leave some required fields incomplete) to review the error messages. All **error handling and error suggestions** should be considered instructions and cues necessary to complete the form. This is regardless of whether the error notification is provided immediately after incorrect input or after submission. If error messages and alerts receive focus and include specific guidance to make necessary corrections, then error instructions and cues do not need to be programmatically associated with a form field.
6	Hidden and disabled form fields do not receive keyboard focus and do not require labels. If they become interactive, then they must be labeled. **Read-only form fields** cannot be edited by the user, but they receive keyboard focus and must meet this requirement.
7	A web page that has **multiple form controls** (e.g. buttons) with the same visual label must include unique identifiers that describe their unique purpose. For example, each “Add” button next to items in a list must identify the specific item that would be added when the button is selected.
	Related: All interactive form elements must be keyboard accessible (Test 1.1).

1.2.2. Results

Note: If the application is not web, mark all of these test IDs DNA. If there are no form fields on a web page, this section should be marked DNA for all Test IDs.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. A Web form field has no markup to associate it to its complete instructions and cues [22(n), B11]	Test ID only applies to web applications. [DNA] if there are no form fields.	[NC] if a form field and all of its instructions and cues do not have HTML (‘Label for’ and ‘ID’ are not used or TITLE is not descriptive) or ARIA for association.	[C] if all form fields and their instructions and cues have HTML (‘Label for’ and ‘ID’ are used or TITLE is descriptive) or ARIA for association.
B. If 'Label for' and 'ID' are used but are not valid HTML [22(n), B11]	Test ID only applies to web applications. [DNA] if there are no form fields that use 'Label for' and 'ID'. [DNA] for a form field that fails 1.2.2.A.	[NC] if 'Label for' and 'ID' are not valid code pairs.	[C] if all 'Label for' and 'ID' are valid code pairs.
C. Form instructions are revealed only by mouse over (TITLE) and are not available onscreen [22(n), B11]	Test ID only applies to web applications. [DNA] if form instructions are not provided by mouseover.	[NC] if form instructions are only provided by mouseover.	[C] if all form instructions that are provided by mouseover are available onscreen.
D. An ARIA form's Name property does not contain complete instructions [22(n), B11]	Test ID only applies to web applications. [DNA] if there are no form fields that use ARIA for markup.	[NC] if an ARIA form field's NAME property does not contain all of its instructions.	[C] if all ARIA form fields have a NAME property that contains all of the instructions.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
E. An ARIA form's Role or State or Value property is not correct [22(n), B11]	Test ID only applies to web applications. Test ID always applies when 1.2.2.D applies. [DNA] if there are no form fields that use ARIA for markup.	[NC] if an ARIA form field's Role, State, or Value properties are not correct.	[C] if all ARIA form fields have correct Role, State, and Value properties.
F. A form control does not identify its purpose [22(n), B11]	Test ID only applies to web applications. [DNA] if there are no form controls.	[NC] if a form control does not identify its purpose.	[C] if all form controls identify their purpose. (Check if there are multiple form controls with the same visual label).

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

22(n) Labels for Forms

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

Applicable Baseline Requirements

B11. Forms (associated instructions)

Labels, instructions, directions and cues necessary to complete a form must be programmatically associated with their respective input control.

1.2.3. Web: Links and User Controls

Links and/or user controls must have meaningful names that describe the unique destination, function, and/or purpose of the control for assistive technology.

Scripting Languages (e.g., JavaScript) can be used to attach a function to an element (image, link, etc.), and display information when triggered by an event (usually a mouse event - 'on mouse over', 'on mouse click', etc.). The scripted element must have a meaningful name.

How to Test	
1	**Use WAT (Doc Info - List Links)** to display a separate page with all links and their attributes (if any). Framed content may need to be tested outside of the frame for this and other WAT tools. Use WAT (Frames – Navigate to framed documents) and then run the link test.
2	**On the links list, use WAT (Doc Info – JavaScript / New Window Links)** to identify which links have a script attached. These links will be marked with a red outline and "link contains JavaScript…" Do not include the "close window" link at the top of the page, which is part of the WAT links list.
3	Review each link under the Link Content column for a **meaningful and unique description that includes destination, function and or purpose of the link**. An image's ALT will display beneath the image. The third column displays the TITLE attribute of the <a> link. (To check for additional TITLE attributes, use WAT (Doc Info – Show Titles). The link must be meaningful for a user of the application. For example, a link named "X17-88" may not make sense to a tester, but in reviewing the rest of the page, it may be clear that it would make sense to a typical user of the application. Links such as "learn more", "more…", or "click here" are not meaningful and do not adequately describe their destination. Inspect their TITLE information for more description. The list of links will include all client-side image map hotspots. The hotspots must have meaningful and unique descriptions.
	Related: All interactive elements must be keyboard accessible (Test 1.1).

Note: Refresh (F5) the page to remove WAT markup.

1.2.3. Results

Note: If the application is not a web application, mark these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. A link does not have a unique and meaningful description [31(a), 31(b), B16]	Test ID only applies to web applications. [DNA] if there are no links.	[NC] if a link does not have a unique and meaningful description. **(Note: this will be NC if 1.2.3.B is NC.)**	[C] if all links have a unique and meaningful description.
B. A scripted element does not have a unique and meaningful description [22(l), 31(a), 31(b), B16]	Test ID only applies to web applications. [DNA] if there are no scripted elements.	[NC] if a scripted element does not have a unique and meaningful description. **(Note: this will cause 1.2.3.A to be NC)**	[C] if all scripted elements have a unique and meaningful description.

Enter the test results into the reporting tool.

<u>Applicable 508 Standards 1194.XX</u>

22(l) Functional Test for Scripts

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

31(a) Use without vision

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

31(b) Use with low vision

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

<u>Applicable Baseline Requirements</u>

B16. Links and User controls

Links and/or user controls must have meaningful names that describe the unique destination, function, and/or purpose of the control for assistive technology.

2. Non-Text interface elements

Non-text elements include images, text rendered as an image, graphs, charts, audio files, animations and video files.

2.1. SW: Images

Software images and graphics include most non-interactive elements: static images, charts, diagrams, text rendered as an image, etc.

How to Test

1 **Find all non-interactive elements** (images, text, data tables, grids, etc.). (Interactive elements were tested in Test 1.2.1)

2 **Use Inspect / Java Ferret.** Mouse over images to check the **Name property for an equivalent text description (purpose and function)**. If the image is of text, the Name must match the text displayed by the image.

If it is a decorative image that has no meaning, Name should equal none if found by Inspector/Java Ferret. It is also acceptable for decorative images to be invisible to the testing tool.

In software data tables (includes calendars), the header cells must be identified. When testing the data cells, the Name property must include the text displayed in the cell. It is common for the Name property of a data cell to also include its header information. This is an acceptable method to provide the Name property for a header cell.

3 **Check that each image's (LegacyIAccessible.)Role is correct.**

4 If the images are not interactive, then (LegacyIAccessible.)State may be normal or read only. **Status indicators must convey current status**. This information can be provided through the Name or State property. For example, an image is often used to indicate the sort status of a data table. The image's (LegacyIAccessible.)Name property or (LegacyIAccessible.)State property must include "Sorted A to Z" or similar.

Complete status information is required. If a data table offers multiple sort options (for example, sortable by date, last name, and city), the data table's current status must include which column is the primary sort option and how that column is sorted. It is not required that a single component provide the complete status for a component (e.g., an asterisk can indicate the column that is the primary sort and a down arrow can indicate that it is sorted alphabetically A to Z).

5 Visually check that images used for **controls and status indicators have consistent meaning**. For example, a red star cannot be used to indicate "overdue" and "completed" in the same application.

2.1. Results

Note: If the application is not a software application, mark these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. An image does not have an equivalent text description (Name), correct Role, or correct State [21(d), B7]	Test ID only applies to software applications. [DNA] if there are no images.	[NC] if an image does not have an equivalent text description (NAME), correct Role, or correct State.	[C] if all images have an equivalent text description (NAME), correct Role, and correct State.
B. Any interface element that has multiple statuses does not indicate its current status [21(d), B6]	Test ID only applies to software applications. [DNA] there are no interface elements with multiple statuses.	[NC] if an interface element with multiple statuses does not indicate its current status in NAME Property.	[C] if all interface elements with multiple statuses indicate their current status.
C. A data table's headers are not identified [21(d), B13]	Test ID only applies to software applications. [DNA] if there are no data tables.	[NC] if a data table's header cells are not identified.	[C] if all data tables' header cells are identified.
D. A complex data table cell is not associated to its header(s) [21(d), B14]	Test ID only applies to software applications. [DNA] if there are no complex data tables.	[NC] if a complex data table's data cells are not associated to their headers.	[C] if all complex data tables' data cells are associated to their headers.
E. An image has inconsistent meaning [21(e), B7]	Test ID only applies to software applications. [DNA] if there are no images.	[NC] if an image has inconsistent meaning throughout the application.	[C] if all images have consistent meaning throughout the application.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

21(d) Name, Role, State

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

21(e) Consistent Images

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

Applicable Baseline Requirements

B6. Multi-state components

Components that can change their state must reveal their current state and function to Assistive Technology.

B7. Images

All images must have associated text describing the purpose and/or function of the image. Decorative images do not require a description.

B13. Data Tables (headers)

Column and row header cells of data tables must be programmatically identified.

B14. Data Tables (cell-header association)

Data cells on complex tables must include markup to associate the data cell with the correct heading.

2.2. Web: Images

Web images include interactive images (links, buttons etc.), static images, charts, diagrams, text rendered as an image, etc.

How to Test

1a Verify that each image has a text equivalent description. For Web images, ALT is most often the attribute used to provide a text description. TITLE can also be used. (In this test script, where ALT is indicated, TITLE is also accepted.)

Use WAT (Images – Show Images) to reveal ALT content and **(DocInfo – Show Titles)** to reveal Titles on images.

1b All images marked by WAT MUST have an ALT or TITLE attribute with an equivalent text description.

- The purpose and/or function of a non-decorative image must be provided.
- Images that are used to display meaningful text must have the same text in its ALT description.
- Decorative images that do not convey any meaning must have ALT="".

If an image is described within page content, it is acceptable for that image to have ALT="".

1c Execute the ARIA Favelet to **determine if ARIA attributes are on the page**. If ARIA is used to describe an image, determine if the text description accurately describes the image's purpose and/or function. The ARIA attribute may contain the text description (aria-label) or reference text on the page.

1d Only images that are marked by WAT or the ARIA favelet are tested in this section. If there are images that are not marked up by WAT or the ARIA favelet:

- They may be background images placed on the page by CSS. (See Test Process 11.)
- They may not be Web images. See the test for software images. (See Test 2.1)

2 If the image is a CAPTCHA, the purpose of the image must be provided. It is acceptable for the CAPTCHA image's ALT="CAPTCHA image of distorted text for visual users". Check that the CAPTCHA meets all other applicable standards (keyboard accessibility, use without vision, use with low vision, etc.)

3 For charts and other graphical representations of data:

- If the chart is intended only to show a trend, ALT content should contain a description of the trend.
- Complex charts and graphs may not be able to be described completely within an ALT attribute. It is acceptable to provide a compliant data table with the

	chart's source data as the equivalent description. It is then acceptable for the chart image ALT to be the name of the chart or similar
4	**Review images that provide status information.** Complete descriptions of the status (opened/closed, expanded/collapsed, sort order, etc.) must be provided. While this information is often provided through images, status may be provided through other means such as text. Identify all multi-state components and check that complete status information is provided. Elements that may be described by ARIA include progress bars and navigation trees.

Note: Refresh (F5) the page to remove WAT markup.

2.2. Results

Note: If the application is not a web application, mark all of these test IDs DNA. Only test Images marked up by the WAT toolbar in this section. (Where ALT is indicated, TITLE is also accepted.)

Failure condition	**Does Not Apply? [DNA]**	**Not Compliant [NC]**	**Compliant [C]**
A. A Web image does not have an ALT or TITLE or ARIA attribute [22(a), B7]	Test ID only applies to web applications. **[DNA]** if there are no images.	**[NC]** if an image does not have an ALT, TITLE, or ARIA attribute. (Do not evaluate an NC image for B, C, or D.)	**[C]** if all images have an ALT, TITLE, or ARIA attribute. (Description does not need to be correct)
B. A meaningful image does not have an equivalent text description (purpose and function) [22(a), B7]	Test ID only applies to web applications. **[DNA]** if there are no meaningful images with ALT or ARIA.	**[NC]** if a meaningful image with ALT does not have an equivalent text description.	**[C]** if all meaningful images with ALT have an equivalent text description.
C. A decorative image does not have ALT="" [22(a), B7]	Test ID only applies to web applications. **[DNA]** if there are no decorative images with ALT or ARIA.	**[NC]** if a decorative image with ALT does not have ALT="".	**[C]** if all decorative images with ALT have ALT="".
D. An image that contains text does not have the same text in the ALT [22(a), B7]	Test ID only applies to web applications. **[DNA]** if there are no images with ALT or ARIA that contain text.	**[NC]** if an image that contains text with ALT does not have the same text in the ALT attribute.	**[C]** if all images that contain text with ALT have the same text in the ALT attribute.

Failure condition	**Does Not Apply? [DNA]**	**Not Compliant [NC]**	**Compliant [C]**
E. A CAPTCHA image ALT does not describe its purpose [22(a), B7]	Test ID only applies to web applications. **[DNA]** if there are no CAPTCHA images.	**[NC]** if a CAPTCHA image does not describe its purpose.	**[C]** if all CAPTCHA images describe their purpose.
F. A multi-state component does not provide complete status information [31(a), 31(b), B6]	Test ID only applies to web applications. **[DNA]** if there are no multi-state components.	**[NC]** if a component that has multiple statuses does not provide its current status.	**[C]** if all components that have multiple statuses provide their current status.
G. An image has inconsistent meaning [21(e), B7]	Test ID only applies to web applications. **[DNA]** if there are no images.	**[NC]** if an image has inconsistent meaning throughout the application.	**[C]** if all images have consistent meaning throughout the application.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

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

22(a) Equivalent Text Descriptions
A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).

31(a) Use without vision
At least one mode of operation and information retrieval that does not require user vision shall be provided, or support for assistive technology used by people who are blind or visually impaired shall be provided.

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

Applicable Baseline Requirements

B6. Multi-state components
Components that can change their state must reveal their current state and function to Assistive Technology.

B7. Images
All images must have associated text describing the purpose and/or function of the image. Decorative images do not require a description.

2.3. Video-only and Animation

Videos (no sound) and animation includes sequences of overlaid images, dynamic changes of state such as a moving speed dial, a chart illustrating dynamic flow changes from one state to another, etc. This requirement applies to visual/video information that is not accompanied by meaningful sounds.

How to Test	
1	Find interface components that play **video-only or animated content** when activated.
2	Manually check that the **information conveyed through animation or video is also available in accessible screen text or audio**. Animation or video must not be the only way to convey information.
3	**The alternative equivalent description must be an accurate and complete representation of the video/animated content.** All visual information must be included (e.g., driver hands license to officer). Equivalent (unsynchronized) audio descriptions for video files are acceptable but not required.
	Related requirements: The placeholder for the video file or animation (typically an image) must meet all applicable standards (Test 2.1 or 2.2). If there is a media player or plug-in, it must meet all applicable standards (Test Process 14).
	Files that have both video and meaningful audio are tested in Test Process 6 Multimedia.

2.3. Results

Note: Do not test multimedia files in this section.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. In SW, an animation does not have an equivalent text or audio description [21(h), B19]	Test ID only applies to software applications. [DNA] if there are no animations.	[NC] if an animation does not have an equivalent text or audio description.	[C] if all animations have equivalent text or audio description.
B. In Web, a video-only file does not have an equivalent text or audio description [22(a), 24(d), B19]	Test ID only applies to web applications. [DNA] if there are no video-only files.	[NC] if a video-only file does not have an equivalent text or audio description.	[C] if all video-only files have equivalent text or audio description.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

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

22(a) Equivalent Text Descriptions
A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).

24(d) Audio Descriptions
All training and informational video and multimedia productions which support the agency's mission, regardless of format, that contain visual information necessary for the comprehension of the content, shall be audio described.

Applicable Baseline Requirements

B19. Video (descriptions)
Video-only files must be accompanied by descriptions.

2.4. Web: Audio-only

Audio-only files include speeches, sound bites, ambient (background) sounds, etc. Video-only files include animations, screen video captures etc.

Note: This test is different from the test for Multimedia (combined audio AND video).

How to Test	
1a	Find interface components that play **audio-only content** when activated
1b	Find the associated transcript or description.
1c	**Find other audio content** that plays automatically or upon activation of a control. **(**Short sounds such as confirmation beeps and error notifications are not included in this requirement.)
2	Find the **associated transcript or description**.
3a	**Play audio and manually check that the information conveyed through audio is also available in accessible screen text/transcript**. Audio must not be the only way to convey information.
3b	**The alternative equivalent description/transcript must be an accurate and complete representation of the audio-only content.** All speech and relevant audio cues must be included (e.g., doorbell rings, car honks, etc.)
	Related requirements: The placeholder for the audio file (typically an image) must meet all applicable standards (Test 2.1 or 2.2). If there is a media player or plug-in, it must meet all applicable standards (Test Process 14).
	Files that have both audio and meaningful video are tested in Test Process 6 Multimedia.

2.4. Results

Note: Do not test multimedia files in this section.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. An audio file does not have an equivalent text description/transcript [22(a), B18]	Test ID only applies to web applications. **[DNA]** if there are no audio only files.	**[NC]** if an audio-only file does not have an equivalent text description/transcript.	**[C]** if all audio-only files have an equivalent text description/transcript.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

22(a) Equivalent Text Descriptions

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

Applicable Baseline Requirements

B18. Audio (transcripts)

Audio-only files must be accompanied by transcripts.

2.5. Web: Image Maps

An image map is a single image that has designated regions or "hotspots" that contain links. Server-side image maps may not be used. Client-side image-maps must be used instead.

How to Test	
1	**Use WAT (Images – Show Image Maps) t**o identify image maps on the page.
2	**Note any server-side image maps found by WAT.**
3	**Manually check** any image maps for keyboard access. If the map's hotspot regions are not keyboard accessible, it is a server-side image map.
	Related requirements: Client-side image map hotspots must have ALT descriptions (Test 2.2) and be keyboard accessible (Test 1.1).

Note: Refresh (F5) the page to remove WAT markup.

2.5. Results

Note: If the application is not a web application, mark all of these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. A server side image map is found [22(e), 22(f), B26]	Test ID only applies to web applications. [DNA] if there are no image maps.	[NC] if a server side image map is found.	[C] if all image maps are client side.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

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

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

Applicable Baseline Requirements

B26. Image maps
Server-side image maps may not be used. Client-side image-maps must be used instead.

3. Color and Contrast

3.1. Color Dependence

Color dependence is using color as the sole means to convey information. For example, a single icon that is green for 'on', orange for 'standby', and red for 'off' is color dependent.

Color must not be the only means of conveying information, indicating an action, prompting a response, or indicating status. Information conveyed through color must also be provided in text on the screen.

How to Test	
1	If there is **information that is being provided by color**, check to see if the same information is represented textually on the screen or through other visual (non-color) differentiation (shape, position, size)
2	Check that the **redundant textual information is visible when the meaningful color is displayed** (i.e., the user should not have to do anything to reveal the text, such as mouse-over to reveal a tool-tip).
3	Determine **software element or web element.**
4a	**Web: Use WAT (Colour - Greyscale)** where content may be color dependent to determine if the meaning is clear when the color is removed. If executing the WAT tool renders a page with missing information, this is not a failure of the Web page. The tool is not functioning properly on this page. Manually inspect the page or print to a black and white printer if necessary.
4b	**SW: Manual inspection**. Print to black and white printer if necessary.

Note: Refresh (F5) the page to remove WAT markup.

3.1. Results

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. In software, information is provided only by color [21(i), B8]	Test ID only applies to software applications. [DNA] if testing a web application. [DNA] if color is not used to provide information.	[NC] if color is the only method used to provide information.	[C] if color is used but is not the only method to provide information.
B. In Web, information is provided only by color [22(c), B8]	Test ID only applies to web applications. [DNA] if testing a SW application. [DNA] if color is not used to provide information.	[NC] if color is the only method used to provide information.	[C] if color is used but is not the only method to provide information.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

21(i) No Color Dependence to Convey Information

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

22(c) No Color Dependence to Convey Information

Pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

Applicable Baseline Requirements

B8. Color (meaning)

Color must not be the only means of conveying information, indicating an action, prompting a response, or indicating status. Information conveyed through color must also be provided in text displayed on the screen.

3.2. Color Contrast

There must be contrasting colors/shades at a minimum ratio of 4.5:1 for discerning between background and foreground content.

How to Test

1 Visually examine **meaningful text and images of meaningful text** on the page for areas that may have low **background to foreground contrast.**

Incidental text, as defined below, is exempt from this requirement:

- part of logos or brand names
- part of an inactive user interface component
- pure decoration
- not visible to anyone
- part of a picture that contains significant other visual content.

All appearances of meaningful text should be evaluated, including changes due to mouse hover and status.

2 **Use WAT (Colour – Contrast Analyser (application))** and use the color picker tool to select foreground and background colors from the screen. Select the Luminosity Algorithm. The Colour Contrast Analyzer (CCA) will display a luminosity contrast ratio, which must be a minimum of **4.5:1**. Use the stand alone CCA or the CCA embedded within WAT for this test.

The WAT contrast analyzer tool works for software and Web testing. If the test workstation utilizes dual monitors, place both the application and the tool in the primary monitor.

3.2. Results

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. The contrast ratio is less than 4.5:1 for content background and foreground colors [31(b), B9]	**[DNA]** if there is no meaningful text.	**[NC]** if the contrast ratio is less than 4.5:1 when comparing the background and foreground colors of meaningful text. (If NC, always document the contrast ratio.)	**[C]** if the contrast ratio is 4.5:1 or greater when comparing all background and foreground colors of meaningful text.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

31(b) Use with low vision

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

Applicable Baseline Requirements

B9. Color (contrast)

There must be contrasting colors/shades at a minimum ratio of 4.5:1 for discerning between background and foreground content.

4. Flashing

The term 'flickering' encompasses interface elements that flash, blink, flicker repetitively, or elements that scroll (e.g., marquee text).

Current Section 508 requirements do not permit elements to flash or flicker at frequencies between 2Hz and 55Hz. To incorporate the expected update to this requirement, this flashing test adopts the WCAG 2.0 frequency restriction to below 3Hz.

DHS-specific test approach:

Due to factors that are not within the developer's control (such as the refresh frequency of a user's monitor, Internet speed, and other display factors), the frequency of a flashing item can be unpredictable from computer to computer. Given this uncertainty, the only specification that can be inspected is the *programmed frequency* of flashing items. Flashing elements that do not have a frequency property or attribute will be found not compliant.

How to Test

1 Visually check for any **flashing or blinking interface elements or scrolling text**. Scrolling images should be tested under Test 2.3 (Video-only and animation).

2 Determine if the flickering element is Software or Web.

3 **Determine the programmatic frequency of the flickering element.** This will often require viewing source code and/or contacting the developer.

To view the source of a part of the page, highlight the section of the page using the mouse. **Use WAT (Doc Info - View Partial Source).**

4. Results

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. The frequency of a flickering element cannot be determined programmatically [21(k), 22(j), B10]	[DNA] if there are no elements that are flashing, flickering, scrolling, etc.	[NC] if the frequency of a flickering element cannot be determined programmatically.	[C] if the frequency of all flickering elements can be determined programmatically.
B. B. The frequency of a flickering element is programmatically set at or above 3 Hz [21(k), 22(j), B10]	[DNA] if there are no elements that are flashing, flickering, scrolling, etc. [DNA] if the frequency is not programmatically set.	[NC] if the frequency of a flickering element is programmatically set at or above 3Hz.	[C] if the frequency of all flickering elements are programmatically set below 3Hz.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

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

22(j) No flickering Interface components
Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

Applicable Baseline Requirements

B10. Flashing
Sections(s) of the screen should not flash at or above 3Hz.

5. Page titles

Programmatically identify Page Titles.

How to Test

1 Examine the **application's title bar or browser tab** when the page initially loads or after a page refresh (F5). Check that it is a meaningful representation / indication of the content. The title should be in plain language (rather than code). The minimum requirement is the Web site and page description or Software application name and screen description.

2 **Some software may not use the title bar**. It may be necessary to use **Alt+TAB** to cycle through open applications, or look in the Windows Taskbar.

5. Results

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. There is no meaningful page title in plain language [31(a), 31(b), B12]	Test ID always applies. **[DNA] is not an acceptable result**.	**[NC]** if there is no meaningful page title in plain language.	**[C]** if there is a meaningful page title in plain language.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

31(a) Use without vision
At least one mode of operation and information retrieval that does not require user vision shall be provided, or support for assistive technology used by people who are blind or visually impaired shall be provided.

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

Applicable Baseline Requirements

B12. Page Titles
Programmatically identify Page Titles.

6. Multimedia

Multimedia is a presentation consisting of time-synchronized media (primarily video and audio).

How to Test

1 **Find interface components that play synchronized media when activated.** This includes streaming media, and streaming live events.

Include only files that have both meaning visual information that is synchronized with meaningful audio information. Examples include webcasts, press conferences, instructional videos, and online training presentations.

If meaningful audio plays while a static image is shown, this should be tested as audio-only (Test 2.4). If meaningful video plays with white noise, this file should be considered video-only (Test 2.3).

2a **Enable captions** in the multimedia player. **Play the multimedia.**

2b **Check captions:** All relevant audible information (dialog and sounds) from the multimedia should be complete, accurately displayed and synchronized with the multimedia audio.

3a **Enable the audio description** feature in the multimedia player. **Play the multimedia.**

3b **Check audio descriptions:** All relevant visual information from the multimedia should be complete, verbally described accurately and synchronized with the multimedia video.

4 **Synchronized alternatives are required.** Transcripts and separate lists of visual events do not meet this requirement.

Note: It is acceptable to have separate files for the captioned version and audio described version of the multimedia.

Related requirements: The multimedia player/plugin must meet all applicable software requirements (Test 14).

6. Results

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. Synchronized captions are not provided for multimedia [22(b1), 24(c), B20]	[DNA] if there is no multimedia file.	[NC] if a multimedia file does not provide synchronized captions.	[C] if all multimedia files provide synchronized captions.
B. The provided captions for multimedia are not equivalent [22(b1), 24(c), B20]	[DNA] if there is no multimedia file. [DNA] if captions are not provided.	[NC] if the provided captions for a multimedia file are not equivalent.	[C] if the provided captions for all multimedia files are equivalent.
C. Synchronized audio descriptions are not provided for multimedia [22(b2), 24(d), B21]	[DNA] if there is no multimedia file.	[NC] if a multimedia file does not provide synchronized audio descriptions.	[C] if all multimedia files provide synchronized audio descriptions
D. The provided audio descriptions are not equivalent [22(b2), 24(d), B21]	[DNA] if there is no multimedia file. [DNA] if audio descriptions are not provided.	[NC] if the provided audio descriptions for a multimedia file are not equivalent.	[C] if the provided audio descriptions for all multimedia files are equivalent.

Enter the test results into the reporting tool.

Note: Separation of 22(b) to b1 and b2 is for reporting of disabilities impacted

Applicable 508 Standards 1194.XX

22(b) Synchronized Multimedia Alternatives
Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.
(Reporting: b1 = captions; b2 = descriptions)

24(c): Synchronized Captions
All training and informational video and multimedia productions which support the agency's mission, regardless of format, that contain speech or other audio information necessary for the comprehension of the content, shall be open or closed captioned.

24(d): Audio Descriptions
All training and informational video and multimedia productions which support the agency's mission, regardless of format, that contain visual information necessary for the comprehension of the content, shall be audio described.

Applicable Baseline Requirements

B20. Synchronized media (captions)
Synchronized media must have captions that are time-synchronized with the dialog and relevant sounds.

B21. Synchronized media (descriptions)
Synchronized media must have audio descriptions that are time-synchronized with the video.

7. Time Outs

Messages and/or instructions to the user requesting their response within a given time are typically associated with sites that require a secure login. This includes both server time-outs and client-side security time-outs.

How to Test

1 **Manual inspection.** Determine if there is a time out function from the application's documentation, or by leaving the session inactive for a period.

Note: Web applications with Flash or embedded Java content must be tested in IE11.

2 **If a time out is about to occur,** an alert must be posted for at least 20 seconds and the user must have the option to request more time.

Related: The alert (often a pop up window) and option to request more time must be keyboard accessible (Test 1.1)

7. Results

Note: If application has no time out feature, mark all test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. The application timed out without notification [22(p), B25]	[DNA] if application does not time out.	[NC] if application times out without notification.	[C] if application provides notification before timing out
B. The application's time out notification is displayed for less than 20 seconds [22(p), B25]	[DNA] if application does not time out. [DNA] if no time out notification is provided.	[NC] if application's time out notification is displayed for less than 20 seconds before timing out.	[C] if application's time out notification is displayed for at least 20 seconds before timing out.
C. The application timed out without an option to request more time [22(p), B25]	Test ID always applies when 7.A applies. [DNA] if application does not time out.	[NC] if application timed out without providing user an option to request more time.	[C] if application provides user an option to request more time before timing out.

Enter the test results into the reporting tool.

Applicable 508 Standards 1194.XX

22(p) Time-Out Notification

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

Applicable Baseline Requirements

B25. Time outs

Users of assistive technology must be alerted about a pending time out, and users must be able to request more time to complete their task.

8. SW: Built-in Accessibility Features

Operating System (OS) accessibility features are applications that are separate to the software under test. The features are usually found and activated in the accessibility (or 'Ease of Access') control panel in Windows.

User-configurable OS accessibility appearance settings and functions must not be disrupted or disabled by the software application.

Note: The navigation to access these features may differ depending on OS updates.

How to Test

1 **Close the stand-alone application or browser (if SW is embedded in a Web page).**

Note: Web applications with Flash or embedded Java content must be tested in IE11.

2 **Enable Windows 7/8.1/10 OS accessibility features as follows:**

2a Set the OS display to **High Contrast Black**: (press Left Alt + Left Shift + Print Screen keys).

(If the software's original color palette is similar to high contrast black, select high contrast white by using the Ease of Access Center.

2b Set **Sticky Keys** to on (press the left Shift key 5 times). An icon will show in the taskbar notification area to show that Sticky Keys is on. (To avoid accidentally disabling Sticky Keys during testing: Press [Windows key + U] to open the Windows Ease of Access Center. Select ‘Make the keyboard easier to use’ and ‘Set up Sticky Keys.’ Under Options, uncheck 'Turn off Sticky Keys when two keys are pressed at once', and under Feedback, check ‘Play a sound when modifier keys are pressed’. Select ‘Apply’ or ‘OK’.)

2c Set **Sound Sentry** to on:

- Press [Windows key + U] to open the Windows Ease of Access center.
- Select ‘Use text or visual alternatives for sounds’, then 'Turn on visual notification for sounds (Sound Sentry)’.
- Select ‘Flash active window’ and ‘Apply’ changes. Press the Shift key and the active window should flash.

2d **If the software is a stand-alone application**:

- Press [Windows key + U] to open the Windows Ease of Access center.
- Select ‘Make the computer easier to see'.
- Select 'Change the size of text and icons'.

- Windows 7: Select 'Set custom text size (DPI)' (in the left navigation) and **set to 200%.**
- Windows 8.1/10: In the ‘Change only the text size’ section, select Menu, and then select a font size twice the current size. Do the same for title bars, message boxes, icons, tooltips, and palette titles.
- Log out and back in if prompted.

3 **Restart the application.**

4 If the software is embedded in a Web page, set the browser zoom control to 200% of normal size. Refresh (F5).

5a Check that the application **adopted the high contrast appearance.**

DHS-specific test: If high contrast is not adopted, check if the application offers its own color and contrast options.

5b Check that the **text size has increased and that information is not cut off** because of the larger font size (scrolling or enlargement of the viewing area may be necessary, and is acceptable).

5c Verify that **Sticky Keys** has not been disrupted in the OS (the icon in the taskbar notification area should be showing that Sticky Keys is still on). Open the Start menu, in the search field, press the SHIFT key, then press the ‘s’ key. A capital letter S should appear.

5d Verify that **Sticky Keys** has not been disrupted in the application. Find any text field in the application, and type a mix of upper and lower case characters using one finger only.

5e Verify that **Sound Sentry** has not been disrupted in the OS. Open Notepad and type [Control+L]. This should cause a beep and the Notepad window to flash if Sound Sentry is working properly.

Note: Another way to verify is to repeat or also check in step 5c. The start menu should have flashed when the shift key was pressed if both Sound Sentry and Sticky keys were functioning properly.

Note: After this test is complete, reset the OS Accessibility features to their default settings and restart the application.

8. Results

Note: If the application is not a software application, mark these test IDs DNA.

Failure condition	**Does Not Apply? [DNA]**	**Not Compliant [NC]**	**Compliant [C]**
A. The application disrupted any of the OS Accessibility options (High Contrast, Sticky Keys, Sound Sentry and/or system text size (stand-alone software only)) [21(b), B28]	Test ID only applies to Software Applications. [DNA] if there is no software.	[NC] if the software application disrupts OS high contrast settings, Sticky Keys, Sound Sentry, or text size on the system.	[C] if all OS accessibility options are not disrupted by the application.
B. The application does not adopt the OS high contrast display settings [21(g), B28]	Test ID only applies to Software Applications. Do not test on Web only applications. **[DNA] is not an acceptable result if testing software (stand alone or embedded)**	[NC] if the application does not adopt the OS high contrast colors.	[C] if application adopts the OS high contrast colors.
C. Text of application did not enlarge or became illegible when enlarged [21(g), B28]	Test ID only applies to Software Applications. Do not test on Web only applications. **[DNA] is not an acceptable result if you are testing software (stand alone or embedded)**	[NC] if text of the application did not enlarge or became illegible when enlarged.	[C] if all text of the application enlarged and remains legible when enlarged.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
D. The application did not adopt the OS high contrast colors, 21(g), and the application does not offer at least 4 color options [*DHS-Specific:* 21(j), no baseline]	Test ID only applies to Software Applications. Do not test on Web only applications. **[DNA] is not an acceptable result if testing software (stand alone or embedded)**	**[NC]** if the application does not adopt the OS high contrast colors AND does not offer at least 4 color options.	**[C]** if application adopts OS high contrast color options **OR** offers at least 4 color options.
E. Sticky Keys functionality was disrupted in the application [31(f), B28]	Test ID only applies to Software Applications. Do not test on Web only applications. **[DNA]** if application is web only. **[DNA]** if application does not have an input text form field to type into.	**[NC]** if Sticky Keys does not work in the application.	**[C]** if Sticky Keys functionality works in the application.
F. Sound Sentry functionality was disrupted by the application [31(c), B28]	Test ID only applies to software applications. **[DNA]** if application is web only.	**[NC]** if Sound Sentry functionality is disrupted by the application.	**[C]** if Sound Sentry functionality is not disrupted by the application.

Enter test results into the reporting tool.

Applicable 508 Standards 1194.XX

21(b) Built-in Accessibility Features

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

21(g) OS Individual Display Attributes

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

21(j) Variety of Color Selections

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

31(c) Use without hearing

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

31(f) Use with physical limitations

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

Applicable Baseline Requirements

B28. Built-in accessibility features

[SW only] Operating System (OS) user-configurable accessibility appearance settings and functions must not be disrupted or disabled by the software application.

9. Web: Text Properties

9.1. Web: Language

A default language must be programmatically identified for each page and for passages that use a language other than the default.

How to Test	
1	Identify the default language of the page and any passages that differ from the default language.
2	**Use the WAT (Doc Info - Show Lang Attributes)** to reveal the language settings applied to the page and to sections of the page.
3	Check that there is an **accurate default language attribute** set for the page.
4	Check that there is an accurate language attribute set **for each passage that is different** from the page's default language.
5	Check that the **language attributes match the actual language used**. Exceptions: proper names, technical terms, or foreign words that have become part of the common language. WAT markup will be at the top of the page and will typically show [html:en] for English. A pop up will display if there is no language attribute found.
	Note: Interface components that are part of the browser and/or operating system, such as dialog boxes, browser status bar, browser title bar, browser menus, are not included in this test (they are outside of the control of the Web page and its language settings).

9.1. Results

Note: If the application is not a web application, mark all of these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. The correct default language for the page is not programmatically set [31(a), 31(b), B17]	Test ID only applies to web applications. [DNA] if the application is a software app.	[NC] if the correct default language for the page is not programmatically set.	[C] if the correct default language for the page is programmatically set.
B. A passage (content, image descriptions, form labels etc.) that differs from the default language of the page is not programmatically identified correctly [31(a), 31(b), B17]	Test ID only applies to web applications. [DNA] if there is not a passage in a language that differs from the default language of the page.	[NC] if a passage on the web page differs from the default language and is not programmatically identified.	[C] if passages on the web page differ from the default language and are programmatically identified.

Enter test results into the reporting tool.

Applicable 508 Standards 1194.XX

31(a) Use without Vision

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

31(b) Use with Low Vision

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

Applicable Baseline Requirements

B17. Language

[Web only] A default language must be programmatically identified for each page, and for passages that use a language other than the default. Exceptions: proper names, technical terms, or foreign words that have become part of the vernacular.

9.2. Web: Section Headings

Headings must be programmatically identified and must match the visual outline level.

How to Test

1a **Check that headings are programmatically marked.** Visually identify where headings are used on the page, through formatting or use of white space, boxes or other visual separators.

1b **Use the WAT (Structure - Headings)** to reveal headings.

1c Check that any visual headings are also **marked as headings**, <H1>, <H2> etc. (If a visual heading is not present, do not make suggestions to add a heading or alter page structure.)

2a **If there are at least 2 visual headings on the page, check that the programmatic heading <H> level structure is correct.** If there are no visual headings or only one visual heading, do not perform this test since there is no heading structure.

2b Examine the programmatic headings of the page. **Use WAT (Structure - Heading Structure) to reveal the <H> hierarchy.** Compare only the programmatic <H> levels to the visual heading level structure on the page. Check that section headings have a more important <H> level than their subtopics (e.g., <H1> is more important than <H2>).

Related: Text in section headings must meet color contrast requirement (Test 3.2)

9.2. Results

Note: If the application is not a web application, mark all of these test IDs DNA

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. Visually apparent headings are not programmatically identified [31(a), 31(b), B15]	Test ID only applies to web applications. [DNA] if there are no visually apparent headings on the page.	[NC] if a visually apparent heading is not programmatically identified with an <H#>.	[C] if all visually apparent headings are programmatically identified with an <H>. (<H> levels do not need to be correct.)
B. Programmatically identified heading levels do not match the visual outline level [31(a), 31(b), B15]	Test ID only applies to web applications. [DNA] if there are no visually apparent headings on the page. [DNA] if there is only 1 visually apparent heading on the page.	[NC] if programmatic <H> levels for a visually apparent heading does not match the visual structure.	[C] if programmatic <H> levels on all visually apparent headings match the visual structure.

Enter test results into the reporting tool.

Applicable 508 Standards 1194.XX

31(a) Use without Vision
At least one mode of operation and information retrieval that does not require user vision shall be provided, or support for assistive technology used by people who are blind or visually impaired shall be provided.

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

Applicable Baseline Requirements

B15. Headings
[Web only] Programmatically identify headings.

10. Web: Data Tables

Data tables are those tables where the information in a cell requires a row or column header to adequately describe the cell's contents.

How to Test

1 **Find all data tables.**

2a **Identify all of the headers for each data cell.** (Some cell headers may not be in the same row and/or column as the data cell.)

2b Determine if a data table is **simple or complex**.

Simple data table: only one level of column headers and/or one level of row headers.

Complex data table:

- Any data table with more than one level of headers is a complex data table.
- If a cell's headers are not in the same row and/or column

3a **Use WAT (Tables – Show Data Tables)**. The tool will mark all tables on the pages. Do not evaluate tables that are used for layout purposes.

3b Table **headers and cells must be within the same table.**

3c For all data tables, **inspect each row and column header for 'SCOPE' or 'ID'.**

Each row and column header must have either SCOPE="row | col", SCOPE="rowgroup | colgroup", or ID="x". If 'ID' is used, data cells must refer to the associated header cell's 'ID' in order for the row and column headers to be properly identified. The IDs must also be valid HTML.

- In HTML 4.01:
    - A valid ID must start with a letter (starting with numbers is prohibited).
    - After the first letter any combination of letters (a to z, A to Z), digits (0 to 9), hyphens (-), underscores (_), colons (:) and periods (.) is allowed.
- In HTML5:
    - A valid ID
        - Must be unique
        - Cannot be the empty string (“”)
        - Cannot contain spaces

3d Only data tables with one or two header levels that are in the same row and/or column as the data cell may use the **'SCOPE' attribute.**

Data tables with **more than two header levels**, or with headers that are not in the same row and/or column as the data cell, must use 'ID' instead of 'SCOPE'. ('ID' can be used on a data table but must be accompanied by the 'headers' attribute in the data cells.)

3e **HTML5** supports TH SCOPE but **does not support TD SCOPE**.

Press F12 or view source to inspect the DOCTYPE.

- <!DOCTYPE HTML> means the page is coded in HTML5. If TD SCOPE is used, it is not valid markup.
- <!DOCTYPE HTML PUBLIC … > means the page is coded in HTML 4.01, and TD SCOPE is valid.

4 Inspect **complex data tables for markup that associates a cell to its headers.**

For complex tables with two header levels, 'SCOPE' is acceptable.

Data tables with more than two header levels, or with headers that are not in the same row and/or column as the data cell, must use 'ID' and 'headers'. Each data cell must have, for example a data table with three levels of column headers, a 'TD HEADERS="x y z"', where "x y z" refers to the correctly associated column headers ('TH ID="x"' on one header and 'ID="y"' on the second header and 'ID="z"' on the third header). If there were row headers, they would also need to be identified and associated.

Any data table that uses 'ID' to identify a header cell must have the 'headers' attribute in its data cells.

5 **Data tables must be rendered with HTML tags**. Using style to modify the appearance of headers and images of data tables are not acceptable.

6 Note: A calendar with days of the week as column headers may be treated as a data table. The visual information must be provided programmatically for each element of the calendar (year, day, month, blackout days, etc.). If all information is provided with the cell data (through TITLE for example), then it may be unnecessary to identify column headers programmatically.

Related: Sort functions, if available, must be keyboard accessible (Test 1.1). Data table must reveal current sorted status (often through an image, Test 2.2)

Note: Refresh (F5) the page to remove WAT markup.

10. Results

Note: If the application is not a web application, mark all of these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. Any data table's row or column headers are not identified programmatically [22(g), B13]	Test ID only applies to web applications. Test ID applies to all data tables (simple and complex). Test ID always applies when 10.C applies. **[DNA]** if there are no data tables or only images of data tables.	**[NC]** if an HTML data table's row or column header is not correctly identified programmatically with valid HTML.	**[C]** if all HTML data tables' row and/or column headers are correctly identified programmatically with valid HTML.
B. An image of a data table is found [22(g), B13]	Test ID only applies to web applications. Test ID always applies when 10.A applies. Test ID always applies when 10.D applies. **[DNA]** if there are no data tables.	**[NC]** if an image of a data table is found.	**[C]** if there are data tables but none of them are images.
C. A complex data table's data cells are not associated with its headers programmatically [22(h), B14]	Test ID only applies to web applications. **[DNA]** if there are no complex data tables or only images of complex data tables.	**[NC]** if an HTML complex data table's data cells are not associated with their headers programmatically.	**[C]** if all HTML complex data tables' data cells are associated with their headers programmatically.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
D. An image of a complex data table is found [22(h), B14]	Test ID only applies to web applications. Test ID always applies when there are complex data tables. Test ID always applies when 10.C applies. [DNA] if there are no complex data tables.	[NC] if an image of a complex data table is found. **(Note: this will cause 10.B to be NC.)**	[C] if there are complex data tables but none of them are images.

Enter test results into the reporting tool.

Applicable 508 Standards 1194.XX

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

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

Applicable Baseline Requirements

B13. Data tables (headers)
Column and row header cells of data tables must be programmatically identified

B14. Data tables (cell-header association)
Data cells on complex tables must include markup to associate the data cell with the correct heading.

11. Web: Style sheet Dependence

Style sheets are a means to provide visual formatting information to complement a web page's content.

How to Test

1 Look at the content on the original page to determine the logical order. Identify any hidden content that is revealed by user interactions.

2 Use WAT (IE - Toggle CSS) to disable style sheets.

3a Visually inspect the page for a **reading order** consistent with style sheet view.

3b Identify any **relevant content (including meaningful images) that is not available without the style sheets.** Check that hidden content is available.

3c Note any **unintended content** that is revealed that could cause confusion, e.g. programming code, overlapping information, etc.

3d Decorative images that are removed do not need to be described. Images that display without CSS should be tested per Test 2.2.

Note: WAT: Toggle CSS to re-enable style sheets. WAF: Refresh (F5) the page.

11. Results

Note: If the application is not a web application, mark all of these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. The order of the content changed and is not logical [22(d), B22]	Test ID only applies to web applications. **[DNA]** if there are no style sheets used.	**[NC]** if the order of the content changed and is not logical.	**[C]** if the order of the content did not change **OR** if the order of the content changed but is still logical.
B. Relevant content or information from a meaningful image is not available [22(d), B22]	Test ID only applies to web applications. Test ID always applies when style sheets are used. **[DNA]** if there are no style sheets used.	**[NC]** if relevant content or information from a meaningful image is not available.	**[C]** if all relevant content and information from all meaningful images is available.
C. Any content overlaps or becomes illegible [22(d), B22]	Test ID only applies to web applications. Test ID always applies when style sheets are being used. **[DNA]** if there are no style sheets used.	**[NC]** if any content overlaps or becomes illegible.	**[C]** if all content does not overlap and stays legible.
D. Programming code or other confusing elements are revealed on the page [22(d), B22]	Test ID only applies to web applications. Test ID always applies when style sheets are used. **[DNA]** if there are no style sheets used.	**[NC]** if programming code or other confusing elements are revealed on the page.	**[C]** if no confusing elements are revealed on the page.

Enter test results into the reporting tool.

Applicable 508 Standards 1194.XX

22(d) Readable without Style Sheets

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

Applicable Baseline Requirements

B22. Style-sheet non-dependence

[Web only] Pages must be structured so that their reading order is consistent whether they are viewed with or without an associated style sheet. Layout and appearance of the page may change visually, as long as the logical reading order is maintained.

12. Web: Frames

Frames are a means of separating out sections of a web page into different navigable regions.

How to Test

1 **Use WAT (Frames – Frame Name / Title). If WAT does not work correctly, use the Frames favelet.** Check each frame for a meaningful and unique content description in Name or Title attribute. Descriptions should be in plain language.

2 Include **iframes** in this test.

WAT Note for testing the content within a frame: WAT functions may not be able test content of multiple frames. Open each framed source separately. To open the source in a separate window, use WAT – Frames – Navigate to framed documents and select the source.

WAF Note for testing the content within a frame: As favelets operate in a more limited context than a toolbar, WAF may not have access to cross-domain content. For example, if an iframe references content from a different domain, WAF will not mark up that content. In these cases, try to navigate to the source of the content and run the relevant tests there.

Note: On framed pages with dynamically generated content, WAF may list a different set of frames than WAT. If you need to load the content of some frames separately for testing purposes and are unable to do so using the list returned by WAF, try using WAT's Navigate to Frames function in IE11.

Related requirement: Each frame must be keyboard accessible (Test 1.1). Use [Control+Tab] or [F6] to move focus through each frame. (This keystroke can vary from application to application. Check for alternative documented keystrokes.)

Note: Refresh (F5) the page to remove WAT markup.

12. Results

Note: If the application is not a web application, mark all of these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. A frame's Title or Name is non-descriptive [22(i), B23]	Test ID only applies to web applications. [DNA] if there are no frames.	[NC] if a frame's Title or Name is non-descriptive.	[C] if all frames' Title or Name is descriptive.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
B. An iframe's Title or Name is non-descriptive [22(i), B23]	Test ID only applies to web applications. **[DNA]** if there are no iframes.	**[NC]** if an iframe's Title or Name is non-descriptive.	**[C]** if all iframes' Title or Name is descriptive.

Enter test results into the reporting tool.

Applicable 508 Standards 1194.XX

22(i) Descriptive Frame Titles

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

Applicable Baseline Requirements

B23. Frames

[Web only] Frames and iframes must have a meaningful description using the title or name attribute.

13. Web: Repetitive Content and Links

A method must be provided to skip blocks of repeated content or links on Web pages allowing a user to move directly to page-specific content.

Groups of links are usually provided along the top and/or left of pages to provide quick navigation to other areas of an application. Similarly, other content may be repeated throughout the site.

To enable equitable use by keyboard only users, there must be a method to skip past repetitive content. On Web pages, this can be provided by methods such as adding internal links that allow a user to move their focus to another part of the page. Similarly, for screen reader users, if they must read content that is repeated on each page and cannot skip past it, their experience on the page can be very frustrating.

How to Test

1 Find **repeated blocks of content and/or repetitive navigation links/menus.**

2 Where components are repeated on multiple pages, **note any changes to the relative order of the repeated components**. If the application has no repetitive Web content, testing is complete.

3a **Web: Locate the skip link(s). Use WAT (Doc Info – Skip Link)** to identify the locations of the links (and #targets) on the page.

If users can navigate frame by frame and effectively skip over the repeated content, a separate skip function is not required. The standard keystroke to navigate to each frame in IE11 and Firefox is <F6>. Chrome does not have a standard frame navigation keystroke.

3b Check that the skip link identifies a **target that is located after the repetitive content.** The testing tools will identify the #targets on the page. If the skip targets are not shown in the expected locations, determine the skip target location with the subsequent steps. If no skip links or targets are revealed, TAB toward the repetitive blocks of content/navigation links to find a skip function. (The skip function may only reveal with keyboard focus so do not rely solely on a visual check.)

4a Refresh (F5) the page and **test the skip link's functionality.**

4b **Tab to the skip function.**

The skip function must be keyboard accessible and receive a visual focus.

4c **Select the function with Enter.**

	(Executing the skip link may cause the visual focus to disappear until the next step. This is not a failure of 21c.)
4d	Tab again. (This step is necessary to determine whether the skip link functions properly.)
4e	The **focus should move beyond** the identified repetitive content to an interactive element (often a link), if there is one. If there is not an interactive element that receives focus, confirm that the target of the skip link is correct. Activate the skip function again. If the target is beyond the visible part of the page, the page will scroll to display the target onscreen. If the page did not scroll, reduce the height of the browser window so that the expected target location is below the visible part of the page, and try again.
5	**Test each skip function.** Each block of repetitive content requires its own skip link. For example, a page with repetitive links should have a skip link to jump past these links. If there is also repetitive content, this should have a separate skip link.
	Related: The skip function must be keyboard accessible (Test 1.1) and have a meaningful name (Test 1.2.3).

Note: Refresh (F5) the page to remove WAT markup.

13. Results

Note: If the application is not a web application, mark all of these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. There is no method to skip repetitive content [22(o), B5]	Test ID only applies to web applications. [DNA] if there is no repetitive content.	[NC] if there is no method/link to skip repetitive content.	[C] if there is a method to skip repetitive content. (Skip Navigation function does not need to work for this test step). [C] if the site utilizes frames and the repetitive content is contained in a frame separate from the main content.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
B. The skip function's target is not located after the repetitive content [22(o), B5]	Test ID only applies to web applications. [DNA] if 13.A is C because the site is using frames. [DNA] if there is no repetitive content or there is no skip target.	[NC] if the skip target is not located after the repetitive content.	[C] if the skip target is located after the repetitive content.
C. The skip function does not work properly [22(o), B5]	Test ID only applies to web applications. [DNA] if there is no repetitive content. [DNA] if 13.A or 13.B is NC.	[NC] if the skip link and target exist but the function does not work properly. (Skip Function does not place focus at the top of the main content).	[C] if all skip functions work properly. (Focus is placed at the top of the main content.)
D. The relative order of repeated components is different from other pages [31(a), 31(b), B3]	Test ID only applies to web applications. [DNA] if there is no repetitive content.	[NC] if the relative order of the repeated components is different from other pages.	[C] if the relative order of the repeated components is the same as other pages.

Enter test results into the reporting tool.

Applicable 508 Standards 1194.XX

22(o) Method to Skip Repetitive Links

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

31(a) Use without vision

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

31(b) Use with low vision

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

Applicable Baseline Requirements

B3. Focus (order)

When the sequence of interface components has meaning or requires an order of operation, the focus/TAB order must logically flow with the application/content.

B5. Skip links

[Web only] A method must be provided to skip blocks of repeated content or links on pages, allowing a user to move directly to page-specific content.

14. Web: Required Plug-ins

When content on public-facing pages can be rendered only via plug-ins, or must be viewed with a separate application, a link to obtain the plug-in and/or application must be provided. Intranet sites are typically accessed by computers that already include the necessary applications to view content and therefore, do not require the link to plug-ins.

If the site has restricted network access (e.g., intranet), and all users have the required plug-in, a link to the plug-in is not required. This exception does not apply to public web pages that require a password-protected user account.

How to Test	
1a	Identify any **file types that cannot be viewed in the browser or without a plug-in.**
1b	Use the **WAT (Doc Info - List Downloadable files)** to identify document file types on the page.
1c	**Software elements, such as Flash, often require a plug-in** to render within a browser. Software may have been previously identified in images, forms, controls, etc. Open the context menu (right click) on the software to help identify the type of file in use. (Java applets and media players may not have the right click capability.)
2	**Return to the Web page and check for links to download the plug-ins required to view the files.** Use WAT (Doc Info – List Links) for assistance. The link(s) can connect directly to the plug-in vendor's website or to a page that contains a list of plug-in links for the site.
3	**Test the plug-in for compliance with all applicable requirements**
4a	**If the plug-in is COTS**, provide a separate report on the plug-in. Media players and document viewers are examples of COTS plug-ins.
4b	**If the plug-in is custom developed,** include the test results as part of the application. Ensure that all applicable Software testing is performed.
5	**Test the content rendered by the plug-in.** Include software content results in the application's report. Documents should be tested with a different process and reported separately.
	Related: The link to the plug-in must be meaningful (Test 1.2.3).The plug-in must meet all applicable Software requirements.

14. Results

Note: If the application is not a web application, mark all of these test IDs DNA.

Failure condition	Does Not Apply? [DNA]	Not Compliant [NC]	Compliant [C]
A. In a public site, the link to download a required plug-in is not provided [22(m), B27]	Test ID only applies to web applications. [DNA] if the site is an internal site. [DNA] if there are no plug-ins required.	[NC] if on a public site, the link to download a required plug-in is not provided.	[C] if on a public site, links to download all required plug-ins are provided.
B. The plug-in required to view content is not compliant [21(a)-(l) as applicable, No baseline]	Test ID only applies to web applications. [DNA] if there are no plug-ins required.	[NC] if the plug-in required to view content is not compliant.	[C] if all plug-ins required to view content are compliant.

Enter test results into the reporting tool.

<u>Applicable 508 Standards 1194.XX</u>

22(m) Link to Download Required Plug-in

When a page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §21(a) through (l).

21(a) through (l) All Software Requirements

<u>Applicable Baseline Requirements</u>

B27. Plug-in Links

[Web only] When public-facing pages utilize content delivered via plug-ins, or contain downloadable content that must be opened with a separate application, a link to obtain the plug-in and/or application must be provided.

15. Web: Alternative (Accessible) Version

Alternative versions for accessibility are permitted only when the primary page cannot be made compliant due to technology limitations and business requirements. The alternative page must provide the same information as the primary page.

How to Test

1 **Verify** that the primary page is not compliant.

2 Determine if there are any **Alternate versions** of the primary page provided.

3 **Check** that the alternative page contains equivalent information as the primary page. Compare the content of the primary page/site and the Alternate page/site, noting any information differences. Ensure that pages with dynamic content are also dynamically updated on the accessible version.

4 **Run all applicable tests** for Software and Web on the accessible version and mark the appropriate test results.

15. Results

Note: If the application is not a web application, mark all of these test IDs DNA.

Failure condition	**Does Not Apply? [DNA]**	**Not Compliant [NC]**	**Compliant [C]**
A. An alternative page does not contain equivalent information as the primary page [22(k), B24]	Test ID only applies to web applications. [DNA] if there is only one version of the site.	[NC] if an alternative page does not contain equivalent information as the primary page.	[C] if all alternative pages contain equivalent information as the primary page.
B. The primary site can be made compliant [22(k), No baseline]	Test ID only applies to web applications. [DNA] if there is only one version of the site.	[NC] if the primary site can be made compliant.	[C] if the primary site cannot be made compliant.

Enter test results into the reporting tool.

Applicable 508 Standards 1194.XX

22(k) Text only or Alternative Versions

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

Applicable Baseline Requirements

B24. Alternate pages

[Web only] When the primary page/site cannot be made accessible, an Alternative page/site must contain equivalent and up-to-date content.

Section 4:
Test Process Mapping

The Application Testing Script consists of fifteen main testing sections. Below is a list of each Section 508 standard that is tested in each section. Testers who are familiar with the Section 508 Software and Web standards and testing tools can use this as a reference while testing.

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

Test	Section 508	Baseline test
1.1 Keyboard Access	21 SW(a) Keyboard Accessibility 21 SW(c) Visual Focus 31 FPC(a) Use without Vision 31 FPC(b) Use with Low vision	B1. Keyboard navigation B2. Focus (visible) B3. Focus (order) B4. Focus (revealing hidden content)
1.2.1 SW: All interactive elements	21 SW(d), Name, Role, State 21 SW(f) Input Text 21 SW(l) Forms	B6. Multi-state components B11. Forms (associated instructions) B16. Links and User controls *+ A test with no baseline*
1.2.2 Web: Forms	22 Web(n) Labels for Forms	B11. Forms (associated instructions)
1.2.3 Web: Links and User Controls	22 Web(l) Functional Test for Scripts 31 FPC(a) Use without vision 31 FPC(b) Use with low vision	B16. Links and User controls
2.1 SW: Images	21 SW(d) Name, Role, State 21 SW(e) Consistent Images	B6. Multi-state components B7. Images B13. Data Tables (headers) B14. Data Tables (cell-header association)
2.2 Web: Images	22 Web(a) Equivalent Text Descriptions 31 FPC(a) Use without vision 31 FPC(b) Use with low vision	B6. Multi-state components B7. Images
2.3 Video-only and Animation	21 SW(h) Animation 22 Web(a) Equivalent Text Descriptions 24 Multimedia(d) Audio Descriptions	B19. Video (descriptions)
2.4 Web: Audio-only	22 Web(a) Equivalent Text Descriptions	B18. Audio (transcripts)
2.5 Web: Image Maps	22 Web(e) Server Side Image Maps 22 Web(f) Client Side Image Maps	B26. Image maps

Test	Section 508	Baseline test
3.1 Color Dependence	21 SW(i) No Color Dependence to Convey Information 22 Web(c) No Color Dependence to Convey Information	B8. Color (meaning)
3.2 Color Contrast	31 FPC(b) Use with low vision	B9. Color (contrast)
4 Flashing	21 SW(k) Blinking Objects 22 Web(j) No flickering Interface components	B10. Flashing (reserved)
5 Page Titles	31 FPC(a) Use without vision 31 FPC(b) Use with Low Vision	B12. Page Titles
6 Multimedia	22 Web(b1) Synchronized Multimedia Alternatives (Captions) 22 Web(b2) Synchronized Multimedia Alternatives (Descriptions) 24(c): Synchronized Captions 24(d): Audio Descriptions	B20. Synchronized media (captions) B21. Synchronized media (descriptions)
7 Time Outs	22 Web(p) Time Out Notification	B25. Time outs
8 SW: Built-in Accessibility Features	21 SW(b) Built-in Accessibility Features 21 SW(g) OS Individual Display Attributes 21 SW(j) Variety of Color Selections 31 FPC(c) Use without hearing 31 FPC(f) Use with physical limitations	B28. Built-in accessibility features + *A test with no baseline*
9.1 Web: Language	31 FPC(a) Use without Vision 31 FPC(b) Use with Low Vision	B17. Language
9.2 Web: Section Headings	31 FPC(a) Use without Vision 31 FPC(b) Use with Low Vision	B15. Headings
10 Web: Data Tables	22 Web(g) Data table headers 22 Web(h) Complex data tables	B13. Data tables (headers) B14. Data tables (cell-header association)
11 Web: Style sheet Dependence	22 Web(d) Readable without Style Sheets	B22. Style-sheet non-dependence
12 Web: Frames	22 Web(i) Descriptive Frame Titles	B23. Frames
13 Web: Repetitive Content and Links	22(o) Method to Skip Repetitive Links 31(a) Use without vision 31(b) Use with low vision	B3. Focus (order) B5. Skip links
14 Web: Required Plug-ins	22 Web(m) Link to Download Required Plug-in 21 SW(a) through(l) All Software Requirements	B27. Plug-in Links + *A test with no baseline*
15 Web: Alternative (Accessible) Version	22 Web(k) Text only or Alternative Versions	B24. Alternate pages + *A test with no baseline*

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

Section 508	Test	Baseline test
21 SW(a) Keyboard Accessibility	1.1 Keyboard Access	B1. Keyboard navigation
21 SW(b) Built-in Accessibility Features	8 SW: Built-in Accessibility Features	B28. Built-in accessibility features
21 SW(c) Visual Focus	1.1 Keyboard Access	B2. Focus (visible) B4. Focus (Revealing hidden content)
21 SW(d) Name, Role, State	1.2.1 SW: All interactive elements 2.1 SW: Images	B6. Multi-state components B7. Images B13. Data Tables (headers) B14. Data Tables (cell-header mapping) B16. Links and user controls
21 SW(e) Bitmap images	2.1 SW: Images	B7. Images
21 SW(f) Input text	1.2.1 SW: All interactive elements	B11. Forms (associated instructions)
21 SW(g) OS Individual display attributes	8 SW: Built-in Accessibility Features	B28. Built-in accessibility features
21 SW(h) Animation	2.3 Video-only and Animation	B19. Video (descriptions)
21 SW(i) No color dependence to convey information	3.1 Color Dependence	B8. Color (meaning)
21 SW(j) Variety of color selections	8 SW: Built-in Accessibility Features	Not applicable
21 SW(k) Blinking objects	4 Flashing	B10. Flashing (reserved)
21 SW(l) Forms	1.2.1 SW: All interactive elements	B11. Forms (associated instructions)
22 Web(a) Equivalent text descriptions	2.2 Web: Images 2.3 Video-only and Animation 2.4 Web: Audio-only	B7. Images B18. Audio (transcripts) B19. Video (descriptions)
22 Web(b1) Synchronized Alternatives (Captions) & 22 Web(b2) Synchronized Multimedia Alternatives (Descriptions)	6 Multimedia	B20. Synchronized media (captions) B21. Synchronized media (descriptions)
22 Web(c) No color dependence to convey information	3.1 Color Dependence	B8. Color (meaning)
22 Web(d) Readable Without Style Sheets	11 Web: Style sheet Dependence	B22. Style-sheet non-dependence
22 Web(e) Redundant text links on server-side image maps	2.5 Web: Image Maps	B26. Image maps
22 Web(f) Client side not server side	2.5 Web: Image Maps	B26. Image maps
22 Web(g) Identify row and column headers	10 Web: Data Tables	B13. Data Tables (headers)

Section 508	Test	Baseline test
22 Web(h) Associate Data with Headers	10 Web: Data Tables	B14. Data Tables (cell-header mapping)
22 Web(i) Descriptive Frame Titles	12 Web: Frames	B23. Frames
22 Web(j) No flickering Interface components.	4 Flashing	B10. Flashing (reserved)
22 Web(k) Text only or Alternative versions	15 Web: Alternative (Accessible) Version	B24. Alternate pages
22 Web(l) Functional Text for Scripts	1.2.3 Web: Links and User Controls	B16. Links and user controls
22 Web(m) Plug-ins	14 Web: Required Plug-ins	B27. Plug-in Links
22 Web(n) Labels for forms	1.2.2 Web: Forms	B11. Forms (associated instructions)
22 Web(o) Method to Skip Repetitive Links	13 Web: Repetitive Content and Links	B5. Skip-navigation links
22 Web(p) Time out notification	7 Time Outs	B25. Time outs
24 Multimedia(c) Captions	6 Multimedia	B20. Synchronized media (captions)
24 Multimedia(d) Audio descriptions	2.3 Video-only and Animation 6 Multimedia	B19. Video (descriptions) B21. Synchronized media (descriptions)
31 FPC(a) Use without vision	1.1 Keyboard Access 1.2.3 Web: Links and User Controls 5 Page Titles 9.1 Web: Language 9.2 Web: Section Headings 13 Web: Repetitive Content and Links	B4. Focus (Revealing hidden content) B6. Multi-state components B12. Page Titles B15. Headings B16. Links and user controls B17. Language (Web only)
31 FPC(b) Use with low vision	1.1 Keyboard Access 1.2.3 Web: Links and User Controls 3.2 Color Contrast 5 Page Titles 9.1 Web: Language 9.2 Web: Section Headings 13 Web: Repetitive Content and Links	B4. Focus (Revealing hidden content) B6. Multi-state components B9. Color (contrast) B12. Page Titles B15. Headings B16. Links and user controls B17. Language (Web only)
31 FPC(c) Use without hearing	8 SW: Built-in Accessibility Features	B28. Built-in accessibility features
31 FPC(f) Use with physical limitations	8 SW: Built-in Accessibility Features	B28. Built-in accessibility features

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

Baseline test	Test	Section 508 coverage
B1. Keyboard navigation	1.1 Keyboard Access	21 SW(a): Keyboard Accessibility
B2. Focus (visible)	1.1 Keyboard Access	21 SW(c): Visual Focus
B3. Focus (order)	1.1 Keyboard Access 13 Web: Repetitive Content and Links	31 FPC(a): Use without vision 31 FPC(b): Use with low vision
B4. Focus (Revealing hidden content)	1.1 Keyboard Access	31 FPC(a): Use without vision 31 FPC(b): Use with low vision
B5. Skip-navigation links	13 Web: Repetitive Content and Links	22 Web(o): Method to Skip Repetitive Links
B6. Multi-state components	1.2.1 SW: All interactive elements 2.1 SW: Images 2.2 Web: Images	21 SW(d): Name, Role, State 31 FPC(a): Use without vision 31 FPC(b): Use with low vision
B7. Images	2.1 SW: Images 2.2 Web: Images	21 SW(d): Name, Role, State 21 SW(e): Bitmap images 22 Web(a): Equivalent text descriptions
B8. Color (meaning)	3.1 Color Dependence	21 SW(i): No color dependence to convey information 22 Web(c): No color dependence to convey information
B9. Color (contrast)	3.2 Color Contrast	31 FPC(b): Use with low vision
B10. Flashing (reserved)	4 Flashing	21 SW(k): Blinking objects 22 Web(j): No flickering Interface components.
B11. Forms (associated instructions)	1.2.1 SW: All interactive elements 1.2.2 Web: Forms	21 SW(f): Input text 21 SW(l): Forms 22 Web(n): Labels for forms
B12. Page Titles	5 Page Titles	31 FPC(a): Use without vision 31 FPC(b): Use with low vision
B13. Data Tables (headers)	2.1 SW: Images 10 Web: Data Tables	21 SW(d): Name, Role, State 22 Web(g): Identify row and column headers
B14. Data Tables (cell-header mapping)	2.1 SW: Images 10 Web: Data Tables	21 SW(d): Name, Role, State 22 Web(h): Associate Data with Headers
B15. Headings	9.2 Web: Section Headings	31 FPC(a): Use without vision 31 FPC(b): Use with low vision
B16. Links and user controls	1.2.1 SW: All interactive elements 1.2.3 Web: Links and User Controls	21 SW(d): Name, Role, State 22 Web(l): Functional Text for Scripts 31 FPC(a): Use without vision 31 FPC(b): Use with low vision
B17. Language	9.1 Web: Language	31 FPC(a): Use without vision 31 FPC(b): Use with low vision

Baseline test	Test	Section 508 coverage
B18. Audio (transcripts)	2.4 Web: Audio-only	22 Web(a): Equivalent text descriptions
B19. Video (descriptions)	2.3 Video-only and Animation	22 Web(a): Equivalent text descriptions 21 SW(h): Animation 24 Multimedia(d): Audio descriptions
B20. Synchronized media (captions)	6 Multimedia	22 Web(b1): Synchronized Alternatives (Captions) 24 Multimedia(c): Captions
B21. Synchronized media (descriptions)	6 Multimedia	22 Web(b2): Synchronized Alternatives (Descriptions) 24 Multimedia(d): Audio descriptions
B22. Style-sheet non-dependence	11 Web: Style sheet Dependence	22 Web(d): Readable Without Style Sheets
B23. Frames	12 Web: Frames	22 Web(i): Descriptive Frame Titles
B24. Alternate pages	15 Web: Alternative (Accessible) Version	22 Web(k): Text only or Alternative versions
B25. Time outs	7 Time Outs	22 Web(p): Time out notification
B26. Image maps	2.5 Web: Image Maps	22 Web(e) Redundant text links on server-side image maps 22 Web(f) Client side not server side
B27. Plug-in Links	14 Web: Required Plug-ins	22 Web(m): Plug-ins
B28. Built-in accessibility features	8 SW: Built-in Accessibility Features	21 SW(b) Built-in Accessibility Features 21 SW(g): OS Individual display attributes 31 FPC(c): Use without hearing 31 FPC(f): Use with physical limitations
No Baseline	1.2.1 SW: All interactive elements 8 SW: Built-in Accessibility Features 14 Web: Required Plug-ins 15 Web: Alternative (Accessible) Version	*Not Applicable*

Document Content Change Log

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

Version 3.06, April 2013

Original published version.

Version 3.07, April 2013

Location	Change
Section 2, Test Environment.	Added “If an application automatically enables compatibility view, do not disable it. Make sure compatibility view is disabled before the testing another application.”
Section 2, Platform, browser, testing tools	Added Named Anchors Bookmarklet
Test 1.2.1. Step 4	Added “(Do not include radio buttons and check boxes in this test.)”
Test Process 8. Step 2c	Added “(To avoid accidently disabling Sticky Keys during testing: Open Ease of Access Center > Set up Sticky Keys > Make the keyboard easier to use > Set up Sticky Keys > uncheck 'Turn off Sticky Keys when two keys are pressed at once.)”
Test Process 8, Step notes	Added to XP instructions “(Settings > uncheck 'Turn off Sticky Keys when two keys are pressed at once)”
Test Process 13, Step 3a	Deleted “and their targets”, replaced with “and #targets”
Test Process 13, Step 3b	Added “For Windows 7, WAT will identify the #targets on the page. For Windows XP, use the Named Anchors Bookmarklet to reveal the #targets on the page.”

Version 3.08, May 2013

Location	Change
Test 1.2.2. Step 4	Added “There are many ways to indicate that a field is required. Usually this is visually indicated by a star (*). This information needs to be directly associated with the input component via one of the above methods (adding a title attribute, including the '*' in the label, or through ARIA ‘required=true’).”
Test 1.2, Failures	Added “If there are no scripted elements, mark [22(l)] as Not Applicable.”

Version 3.1, July 2013

Location	Change
Section 2, Test Environment	Added IE9
Section 2, Web Accessibility Toolbar Settings	Added "Some functions do not work on pages within Frames. Open each page in its own window before running WAT tool functions. Confirm all WAT popup messages by reviewing WAT markup on page."
Section 2, Platform, browser, testing tools	Added Frames Favelet
Section 2: Compliance Tests	Bullets changed to alphabetized items (Test ID) for each Failure
Test 1.2.1, Failure C	Added "interactive"
Test 9.1, Failure A	Added "correct default"
Test 9.1, Failure B	Added "correctly"
Test Process 12, Step 1	Added "If WAT does not work correctly, use the Frames favelet."

Version 3.1.1, July 2013

Location	Change
Test 1.2.2, Step 7 & Failure F	Added test step and associated failure for a web form control not identifying its purpose.

Version 3.2, August 2013

Location	Change
Test 1.1,Step 2c	Corrected cross-reference to TITLE (from 1d to 1e)
Test 1.2.2, Step 2b.	Moved “Fieldset and Legend may be used for grouping and associating two instructions to one input field.” (from 2e to 2b)
Test 1.2.3, Step 3	Changed “Title information is displayed in the third column.” to “The third column displays the TITLE attribute of the <a> link. (To check for additional TITLE attributes, use WAT (Doc Info – Show Titles).”
Test 1.2.3., Failure B	Added “unique and”
Test 2.1, Step 4	Added Sorted A to Z “or similar”
Test 2.5, Note on Steps	Corrected cross reference 2.3 to 2.2
Test Process 5, Step 1	Added “The minimum requirement is the Web site and page description or Software application name and screen description.”
Test Process 6, Step 1	Deleted “Use the WAT (Doc Info - List of Multimedia files) to find multimedia files.”
Test Process 8,	Added “Note: The navigation to access these features may differ depending on OS updates.”
Test Process 8	Added more details to Steps 2c, 2d, 2e, 5c, and 5e test instructions.
Test Process 8, Failure B	Added “OS”
Test Process 11, Failure D	Added “or other confusing elements are”
Test Process 12, Step 1	Added “Descriptions should be in plain language.”
Test Process 13, Step 2 and Failure D	Deleted “software screens”
Section 3: All Tests	Deleted all instructions for entering Compliant and Not Applicable results. Replaced with tables detailing how to enter results.

Version 3.2.1, December 2013

Location	Change
Section 3 and multiple	Added naming convention for Section 3: Test Process Test Process 1 – Interactive Interface Elements (main category) Test 1.1 Keyboard Access (sub-category) Step 1a (instructions) Test ID 1.1.A.(failure condition)
Test 2.2, Step 1c	Added “If ARIA is used to describe an image, determine if the text description accurately describes the image’s purpose and/or function. The ARIA attribute may contain the text description (aria-label) or reference text on the page.” Deleted use of Inspect from this test.
Test 2.3	Corrected reference to Sections 2.1, 2.2 in Related Requirements
Test 3.2, Step 1	Test is for meaningful text and images of meaningful text only. Deleted “Visually examine meaningful text and images of meaningful text on the page for areas that may have low background to foreground contrast”
Test 4, Step 1	Corrected reference to “Section 2.3 (Video-only and animation)”

Version 3.3, February 2014

Location	Change
Web Accessibility Toolbar: Installation advice	Changed URL to point to September 2012 version.
Web Accessibility Toolbar: Current version	Added “Note: This is the current version for use in this Test Process. Later versions of WAT available from the official site (http://wat-c.org/tools/) may not function properly on DHS workstations.”
Which Testing Tool should I use?	Added “If it opens in a browser and WAT does not provide markup on form fields or images, check if it the ARIA favelet marks these elements.”
Test 1.1, Step 1d	Added “Hidden and disabled form fields are not interactive and do not require keyboard access.”
Test 1.1, Step 2a	Added “Visual focus is an indication of the keyboard focus location. Only keyboard accessible elements need to be analyzed for visual focus.”
Test 1.1, Step 2d	Added “Evaluate the existing focus order only; do not consider elements that do not receive keyboard focus.”
Test 1.1, Step 2e	Added “In IE8, visual focus is lost on frames. This is a failure.”
Test 1.2.1, Step 1	Separated user controls (buttons and menus) from form field elements
Test 1.2.2, Step 3a	Added “ARIA (Accessible Rich Internet Applications) defines a method that specifies how to increase the accessibility of web pages, in particular, dynamic content and user interface components.”
Test IDs 2.2.B, 2.2.C, 2.2.D	Added to guidance “with ALT”
Test 2.4	Corrected reference to Sections 2.1, 2.2 in Related Requirements
Test 3.2, Step 1	Added “Include all appearances of text including changes due to mouse hover and status.”
Test 9.2, Steps 2a and 2b	Evaluate programmatic headings only.
Test ID 9.2.B	Removed “[NC] if 9.2.A is NC”
Test Process 10, step 6	Added “with the cell data (through TITLE for example)”
Test IDs 10.A and 10.C	Added to DNA guidance “or only images of [complex] data tables.”

Version 3.4, December 2014

Location	Change
Section 2: Test Environment	Added F12 developer tools instructions for standard view ("Another method to configure…")
Section 2: Test Environment, all tools	Changed URL to Download from URL.
Test 1.2.2, Failure B	Added "[DNA] for a form field that fails 1.2.2.A."
Test 1.2.3, Step 1	Added "Framed content may need to be tested outside of the frame for this and other WAT tools. Use WAT (Frames – Navigate to framed documents) and then run the link test."
Test 2.2, Results	Added to note: "(Where ALT is indicated, TITLE is also accepted.)"
Test 2.2, Failure A	Added to NC "(Do not evaluate an NC image for B, C, or D.)"
Test 2.2, Failure B,C, & D	Added to DNA: "with ALT or ARIA"
Test 3.2, Failure A	Deleted "Test ID always applies. [DNA] is not an acceptable result". Replaced with "[DNA] if there is no text."
Test 5, Step 1	Added "when the page initially loads or after a page refresh (F5)."
Test 13, Failure A	Added "/link" to "clarify "if there is no method/link…"
Test 13, Failure B	Deleted "[NC] if the target of a skip function is not located after the repetitive content". Replaced with "[NC] if there is no skip target." Deleted "[C] if the target of all skip functions is after the repetitive content". Replaced with "[C] if there is a target for all skip links"
Test 13, Failure C	Added to DNA text " [DNA] if 13.A or 13.B is NC."
Test 13, Failure C	Deleted "[NC] if the skip function does not work properly." Replaced with "[NC] if the skip link and target exist but the function does not work properly."

Version 3.4.1, May 2015

Location	Change
Section 2: Test Environment	Added IE11 as an acceptable browser
Section 2: Test Environment	Updated Inspect installation instructions

Version 4.0, April 2017

Location	Change
Title page	Changed name of document from “DHS Section 508 Compliance Test Process …” to “Trusted Tester Section 508 Compliance Test Process …”
Throughout	Added notes regarding testing of Java, Inspect, and ARIA: • Web applications with Flash or embedded Java content should be tested in IE11 to determine the accessibility of the coded content. • Removed “DHS” where applicable.
Section 2: Test Environment	Added Windows 8.1 and 10 as supported OSes for testing. Added Firefox and Chrome as supported browsers for testing. Removed support for Windows XP and IE versions 8, 9. Added note that IE11 is most accessible test environment.
Section 2: Test Environment	Updated configuration instructions for Inspect and Ferret. Added clarification regarding categorization of “software”. Added sections for WAF and Colour Contrast Analyzer to Tools section. Moved installation instructions to tools installation document. Removed Named Anchor Bookmarklet, since XP is no longer supported.
Section 1, Issues that are Not Covered in this Test Process	Removed two examples from “Issues that are not covered in this test process”
Section 2, Platforms, Browsers, Testing Tools	Rearranged content.
Section 2, Testing Tools	Moved tools installation and settings information to Trusted Tester Test Environment Installation and Configuration Guide
Section 2, Testing Tools	Added WAT version 2015. Added Thatcher’s Skip Links favelet.
Various	Removed support for Windows XP and IE versions 8, 9
Test 1.2.1 (SW interactive elements)	Added instructions to use mouse for keyboard-inaccessible elements. Added instructions to check for required field indicators.
Test 1.2.2 (Forms)	Added information about checking HTML version and ID naming in HTML5. Changed test for ARIA elements to use ARIA favelet instead of Inspect. Added check for “required=true”.
Test 3.2 (Color Contrast)	Added clarification that contrast ratio of 4.5:1 is the minimum. Added exemption for incidental text. Results A: Specified that tests apply to non-incidental text only. Applicable Baseline Requirements: Clarification of minimum ratio.
Test 7 (Timeouts)	Added reminder that web applications with Flash or embedded Java content must be tested in IE11.
Test 8 (Built-in Accessibility Features)	Updated instructions to use [Windows key + U] keyboard shortcut for opening Ease of Access center for more consistency. Updated instructions to include testing in Windows 8.1 and Windows 10. Added reminder that web applications with Flash or embedded Java content must be tested in IE11. Updated instructions for triggering sound. Removed Windows XP information.

Location	Change
Test 12 (Web Frames)	Added note for testing content within a frame and possible use of the Navigate to Frames function applies to WAT only (not WAF). Added notes for WAF testing of frames.
Test 13 (Repetitive Content)	Intro paragraph: added “methods such as” to clarify that internal links are just one example of how skip functions may be provided. Note added: F6 does not navigate frames in Chrome, which does not have a standard frame navigation keystroke. Removed instructions for Windows XP (use of Named Anchors Bookmarklet). Added: If the Skip Link tool does not show skip targets in the expected locations, determine the skip target location with the subsequent steps in the testing process (refresh the page, tab to the skip function, activate it using Enter key). Added instructions for if no skip links or targets are revealed. Added instructions for checking where the focus moves after activating the skip function.

DHS Application Testing Process Quick Reference

The DHS Application Testing Process consists of fifteen main testing sections. Below is a list of each Section 508 standard that is tested in each section. Testers who are familiar with the Section 508 Software and Web standards and testing tools can use this as a quick reference while testing.

1 INTERACTIVE INTERFACE ELEMENTS

1.1 Keyboard Access

- SW(a) Keyboard Accessibility*
- SW(c) Visual Focus*
- FPC(a) Use without Vision
- FPC(b) Use with Low vision
- ** Always test for SW a (keyboard) & SW c (focus) for Software and Web.*

1.2.1 SW: All interactive elements

- SW(d) Name, Role, State
- SW(f) Input Text
- SW(l) Forms

1.2.2 Web: Forms

- Web(n) Labels for Forms

1.2.3 Web: Links and User Controls

- Web(l) Functional Test for Scripts
- FPC(a) Use without vision
- FPC(b) Use with low vision

2 NON-TEXT INTERFACE ELEMENTS

2.1 SW: Images

- SW(d) Name, Role, State
- SW(e) Consistent Images

2.2 Web: Images

- Web(a) Equivalent Text Descriptions
- FPC(a) Use without vision
- FPC(b) Use with low vision

2.3 Video-only and Animation

- SW(h) Animation
- Web(a) Equivalent Text Descriptions
- Multimedia(d) Audio Descriptions

2.4 Web: Audio-only

- Web(a) Equivalent Text Descriptions

2.5 Web: Image Maps

- Web(e) Server Side Image Maps
- Web(f) Client Side Image Maps

3 COLOR AND CONTRAST

3.1 Color Dependence

- SW(i) No Color Dependence to Convey Information
- Web(c) No Color Dependence to Convey Information

3.2 Color Contrast

- FPC(b) Use with low vision

4 FLASHING

- SW(k) Blinking Objects
- Web(j) No flickering Interface components

5 PAGE TITLES

- FPC(a) Use without vision
- FPC(b) Use with Low Vision

6 MULTIMEDIA

- Web(b1) Synchronized Multimedia Alternatives (Captions)
- Web(b2) Synchronized Multimedia Alternatives (Descriptions)
- Multimedia(c): Synchronized Captions
- Multimedia(d): Audio Descriptions

7 TIME OUTS

- Web(p) Time Out Notification

8 SW: BUILT-IN ACCESSIBILITY FEATURES

- SW(b) Built-in Accessibility Features
- SW(g) OS Individual Display Attributes
- SW(j) Variety of Color Selections
- FPC(c) Use without hearing
- FPC(f) Use with physical limitations

9 WEB: TEXT PROPERTIES

9.1 Web: Language

- FPC(a) Use without Vision
- FPC(b) Use with Low Vision

9.2 Web: Section Headings

- FPC(a) Use without Vision
- FPC(b) Use with Low Vision

10 WEB: DATA TABLES

- Web(g) Data table headers
- Web(h) Complex data tables

11 WEB: STYLE SHEET DEPENDENCE

- Web(d) Readable without Style Sheets

12 WEB: FRAMES

- Web(i) Descriptive Frame Titles

13 WEB: REPETITIVE CONTENT AND LINKS

- Web(o) Method to Skip Repetitive Links
- FPC(a) Use without vision
- FPC(b) Use with low vision

14 WEB: REQUIRED PLUG-INS

- Web(m) Link to Download Required Plug-in
- SW(a) through(l) All Software Requirements

15 WEB: ALTERNATIVE (ACCESSIBLE) VERSION

- Web(k) Text only or Alternative Version