

AUDIT READINESS BRIEF · SOFTWARE APPLICATIONS

Section 508 Audit Readiness: Software

What a formal Section 508 validation covers for desktop, client-server, and platform-native software applications. Aligned with WCAG 2.0 AA and Chapter 5 Software requirements of the Revised 508 Standards.

READING TIME

10 minutes

AUDIENCE

Software Vendors & Primes

PUBLISHED

April 2026

About this brief

Software applications delivered under federal contract must meet both WCAG 2.0 AA and Chapter 5 Software requirements. Chapter 5 adds obligations that automated web scanners do not evaluate.

Simkins & Elgazar is an independent Section 508 compliance validation firm serving federal contractors and subcontractors. Our methodology is aligned with the DHS Trusted Tester conformance test process, the recognized standard for manual accessibility evaluation of federal electronic and information technology deliverables.

- Independent validation authority with no conflict of interest
- DHS Trusted Tester-aligned evaluation methodology
- Fixed-fee, documentation-first engagement model
- Federal contractor and subcontractor focus
- Structurally independent from remediation implementation

IMPORTANT

Chapter 5 includes additional requirements for interoperability with assistive technology, platform accessibility services, and user controls for captions and audio descriptions. Automated testing tools designed for web content do not evaluate software-specific requirements.

Purpose of this brief

This readiness brief addresses accessibility evaluation considerations specific to software applications delivered under federal contract. Desktop applications, client-server systems, and platform-native applications are subject to Section 508 requirements under Chapter 5 (Software) of the Revised 508 Standards and applicable WCAG 2.0 Level A and AA success criteria. This document does not contain procedural test scripts, scoring methodologies, or pass/fail determination criteria. Formal compliance determination requires structured manual evaluation by a certified accessibility professional.

Regulatory and Enforcement Context

Section 508 of the Rehabilitation Act requires federal agencies to ensure that electronic and information technology, including software applications, is accessible to individuals with disabilities. The applicable standards include WCAG 2.0 Level A and AA success criteria as well as the functional performance criteria and Chapter 5 Software requirements of the Revised 508 Standards (36 CFR Part 1194).

Software applications must support platform accessibility services and interoperate with assistive technologies, including screen readers, screen magnifiers, and alternative input devices. Applications must not disrupt the operation of accessibility features provided by the underlying platform.

Automated testing tools designed for web content do not evaluate software-specific requirements. Manual evaluation using structured methodology and assistive technology verification is required for compliance determination.

Evaluation Domains Overview

Software accessibility evaluation encompasses the WCAG-aligned requirements applicable to web content as well as additional software-specific requirements. The following provides a high-level overview of the primary evaluation areas.

2.1 Keyboard operability and input

All interactive functionality must be operable through keyboard input. Software must not override or interfere with platform keyboard conventions. Focus must be visible, follow a logical order, and never become trapped. Applications must support standard keyboard shortcuts and not require simultaneous key activation for essential functions where the platform provides an accessibility mode.

- Custom controls that do not respond to standard keyboard interactions
- Focus that becomes trapped in dialogs, panels, or embedded components
- Non-standard keyboard behavior that conflicts with platform conventions

2.2 Assistive technology interoperability

Software must expose accessibility information through the platform's accessibility API. This includes the name, role, state, and value of all user interface components. Changes in state must be communicated programmatically to assistive technology. Components must not suppress or override information provided through the accessibility API.

- UI components without programmatic names or roles
- State changes not communicated through the accessibility API
- Custom controls that bypass platform accessibility services

2.3 Forms, controls, and error handling

Form elements and interactive controls must be programmatically associated with descriptive labels. All instructions, constraints, and error messages must be conveyed to assistive technology users. Error prevention mechanisms are required for transactions with legal, financial, or data-modification consequences.

- Controls without programmatic labels or with labels that do not describe the function

- Error messages not associated with the specific control in error
- Missing confirmation or review mechanisms for consequential transactions

2.4 Visual presentation and display

Text must maintain sufficient contrast against its background. Color must not be the sole means of conveying information. Content must be usable when text is resized through platform accessibility settings. The application must not override user-selected display preferences for high contrast mode or text size.

- Insufficient contrast in custom-themed UI elements
- Application UI that becomes unusable in platform high contrast mode
- Text and controls that do not respect platform text scaling settings

2.5 Synchronized media and audio

Software that presents synchronized media must provide accurate captions and, where applicable, audio descriptions. User controls for captions and audio descriptions must be provided at the same menu level as volume and program selection controls. Auto-playing audio must include accessible pause or volume mechanisms.

- Embedded media players without caption or audio description controls
- Caption controls buried in sub-menus below volume or program selection

2.6 Platform accessibility integration

Software must not disrupt platform-level accessibility features, including screen readers, magnification, high contrast modes, and keyboard accessibility settings. Applications must use standard platform controls where possible or implement custom controls that fully support platform accessibility APIs.

- Custom rendering that bypasses platform accessibility drawing APIs
- Application behavior that disables or interferes with platform screen reader functionality

Observed Failure Patterns

The following patterns represent recurring deficiencies observed across federal subcontractor software deliverables.

Custom control accessibility

Custom-developed user interface controls frequently lack programmatic exposure of name, role, state, and value through the platform accessibility API. Tree views, data grids, ribbon interfaces, and custom dialogs are particularly prone to these deficiencies. Controls that function correctly for mouse users may be entirely inoperable for keyboard and assistive technology users.

Platform API integration

Applications that use non-standard rendering engines or custom drawing routines frequently bypass the platform's accessibility services. This results in content that is invisible to screen readers and other assistive technologies, regardless of its visual presentation.

State and notification failures

Applications frequently fail to communicate state changes to assistive technology. Selection state, expanded or collapsed state, enabled or disabled state, and progress indicators are commonly omitted from accessibility API exposure. Modal dialogs and transient notifications are often not announced to screen readers.

High contrast and display adaptation

Applications that implement custom visual themes often fail when platform high contrast mode is enabled. Text may become invisible, borders may disappear, and interactive elements may become indistinguishable from static content. Applications that hard-code font sizes do not scale when users adjust platform text size settings.

Pre-Audit Preparation Considerations

The following considerations may assist subcontractors in reducing accessibility risk prior to formal validation of software deliverables.

Platform accessibility service verification

Use platform-native accessibility inspection tools to verify that all interactive controls expose their name, role, state, and value through the accessibility API. On Windows, tools such as Accessibility Insights and the Inspect tool from the Windows SDK can reveal accessibility properties. Gaps in API exposure are a strong indicator of audit findings.

Assistive technology compatibility

Test critical workflows with the screen reader commonly used in the target environment. Verify that control names are announced, state changes are communicated, and navigation between components functions as expected. Screen reader compatibility testing addresses a category of defects that no automated tool can detect.

Keyboard-only workflow testing

Navigate all primary user workflows using only the keyboard. Verify that every interactive element can be reached, activated, and exited without a pointing device. Pay particular attention to modal dialogs, tree views, data grids, and custom dropdown controls.

Platform display adaptation

Enable platform high contrast mode and verify that all text, borders, icons, and interactive elements remain visible and distinguishable. Adjust platform text size settings and verify that the application layout accommodates larger text without truncation or loss of functionality.

What a Formal Validation Produces

A formal Section 508 compliance validation conducted by Simkins & Elgazar produces structured documentation suitable for submission to prime contractors, contracting officers, and agency Section 508 coordinators.

Structured issue log

Each deficiency is documented with precise location, affected WCAG success criterion or Chapter 5 requirement, severity classification, and a description of the barrier to access.

Evidence documentation

Findings are supported by annotated screenshots, accessibility API inspection output, and detailed reproduction steps. Evidence includes assistive technology interaction results where applicable.

Standards mapping

Each finding is mapped to the applicable WCAG success criterion, Chapter 5 Software requirement, or Functional Performance Criterion, ensuring traceability to the governing standards.

Re-test verification

Following remediation, independent re-test verification confirms that identified deficiencies have been resolved without introducing new barriers.

Formal compliance documentation

The final deliverable package constitutes a formal compliance record suitable for contract deliverable submissions, VPAT/ACR documentation, and agency review.

WHAT NEXT

Get Ahead of the Audit

Software accessibility findings are expensive to fix late in the development cycle. The fastest way to know where your application stands is to have a qualified manual tester walk a primary workflow with a screen reader and keyboard only. We offer a free accessibility scan for organizations that want an independent read before a formal audit.

Schedule a Free Accessibility Scan

We will evaluate a representative workflow in your software application and return a short report with the real findings, mapped to WCAG and Chapter 5 requirements. No vendor pitch. Just the answer.

· hello@simkinselgazar.com

SCHEDULE YOUR FREE SCAN →

PREPARED BY SIMKINS & ELGAZAR · INDEPENDENT SECTION 508 COMPLIANCE VALIDATION ·
SIMKINSELGAZAR.COM