CMS Section 508 IT Developer Toolkit

Approved April 16, 2025

Introduction

The Centers for Medicare and Medicaid Services (CMS) requires all deliverables to be Section 508 accessibility compliant as part of every contract. CMS will validate all applications as Section 508 compliant using defined expectations defined in this toolkit.

Application development organizations (ADOs) should use this supplier toolkit to make their products accessible. This toolkit is a living document subject to changes or enhancements as necessary. Any references to “accessibility” in this toolkit are to be interpreted as Section 508 compliance. You may provide comments about this toolkit by emailing CMS_Section508@cms.hhs.gov.

The following items will undergo Section 508 evaluation:

- User interfaces used by CMS staff, contractors, or members of the public
- All documentation and training related to product development

Section 508 evaluation / testing will be conducted on all final deliverables and services before acceptance and deployment by CMS.

Manual and automated testing will be conducted based on the Web Content Accessibility Guidelines (WCAG) 2.1 accessibility standards. All issues that are determined to be high severity (Severity 1) for user impact should be remediated before release.

Using this toolkit will reduce costs by embedding Section 508 concepts throughout the product lifecycle instead of addressing them later through costly remediation. This toolkit will help ADOs, CMS development teams, and document creation teams, whether internal or external, find a subset of WCAG 2.1 accessibility issues.

This supplier toolkit describes what needs to be considered for compliance. It outlines the tools, methodology, and evaluation factors that will be used in validating Section 508 compliance. CMS uses industry best-practice, open-source tools and assistive technology to measure compliance. Development operations (DevOps) teams or ADOs that apply the toolkit throughout the product lifecycle will have a smoother accessibility test and compliance reporting process.

This toolkit does not interfere with or conflict with usage of the Accessibility Requirements Tool (ART). It supports the requirements that are output from ART and makes it easier to meet those requirements.

The Contracting Officer's Representative (COR) provides this toolkit to contractors upon contract award.

To work effectively with the Section 508 team:

- Read this supplier toolkit.
- Collect and retain results of tool output throughout the development lifecycle. This information will help you communicate effectively with the Section 508 team during consultations and provide the Section 508 team with updates.
- To get help with specific issues, ask for a consultation by emailing CMS_Section508@cms.hhs.gov.

1.0 Documents

1.1 Roles and Responsibilities

Authors are responsible for document compliance. If the document is a contract deliverable and will be converted to another format, the source document must be accessible before conversion. CMS may need to obtain the source document if compliance issues are found.

1.2 Tools

The following tools are required to ensure that documents are accessible. All errors found in accessibility checks must be fixed.

1) Accessibility Checker for Microsoft Word, PowerPoint, Excel
2) Accessibility Checker for Adobe PDF
3) HHS Accessibility Conformance Checklists

1.3 Accessibility Quick Checks

The following additional checks should be done during document creation. At a minimum, the following need to be confirmed before getting help from clearance officers.

1) Microsoft PowerPoint:
   a) Make sure all text appears in outline view.
   b) Note, reading order issues must be fixed per section 1.2 Tools.
2) Microsoft Word:
   a) Make sure there's a perfect heading tree with top level heading and no skipped heading levels.
3) Microsoft Excel:
   a) Make sure there are no blank cells. These cause problems with screen readers.
4) Adobe PDF:
   a) Check reading order for anything unexpected or extra.

        i) Example of extra: page borders.
        ii) Example of unexpected: reading order significantly different from visual order.
    b) Check the tag tree for anything unexpected.
        i) Example of unexpected: element with incorrect type.
5) All document types:
    a) Make sure all figures have alternative text (alt text).
    b) Turn on screen reader. Use the down arrow key to read through the document, and listen for anomalies.
    c) If there’s a form in the document, use the Tab key to make sure all field elements (buttons, edit fields, drop down boxes, etc.) are reachable by keyboard.

2.0 Products

2.1 Roles and Responsibilities

To ensure accessibility development operations (A11yDevOps) run smoothly, you must define several important roles and responsibilities.

The more responsibilities are covered, the more successful Section 508 compliance will be. One person may cover several roles and responsibilities if necessary. Suppliers should have the following:

1) Program manager/Product manager:
    a. Determine if something is new to the industry or complex, and would therefore be a candidate for in-depth design sprints under the Agile development methodology.
    b. Keep track of third-party dependencies for reporting and follow-up.
2) Designer/Researcher:
    a. Choose the right specification types (basic, walkthrough/storyboard).
    b. Use a basic specification for small changes.
    c. Use basic and walkthrough or storyboard specifications for large changes, custom controls, and complex changes.
    d. Conduct research with a variety of low-, medium-, and high-fidelity prototypes (at various levels of completeness).
    e. Review changes, custom controls, and complex changes with subject matter experts (SMEs) or champions.
3) Developer:
    a. Ask designers for accessible designs.
    b. Instead of developing custom controls, ask whether an existing, accessible design system (such as the CMS Design System or the United States Web Design System) may be used.
    c. Include accessibility events in developer specifications.
    d. Ensure the code follows accessibility standards.

    e. Implement continuous improvement/continuous development (CI/CD) in pipeline, using the tools recommended in section 2.2 Plug-ins and Tools.
    f. Run automated and basic manual checks on every code check-in, using the tools recommended in section 2.2 Plug-ins and Tools.
4) Tester:
    a. Run manual scripts and test cases such as Microsoft Accessibility Insights manual pass for web, or use the Access Board ICT Baseline Testing.
    b. Run assistive technology as needed, using the tools recommended in section 2.2 Plug-ins and Tools.
5) Everyone:
    a. Learn to triage accessibility bugs.
    b. Understand user needs.
    c. Participate in accessibility bug bashes.
6) Assistive technology users and people with disabilities:
    a. Participate throughout the product lifecycle as SMEs (particularly in design, research, and test phases).

Note: CMS Section 508 testing will be more efficient and thorough if you test your product with automated tools and get a clean, error-free result before starting manual testing.

2.2 Plug-ins and Tools

We recommend the following plug-ins and tools to check for accessibility at various stages in the lifecycle.

These tools are recommended, but comparable tools and deliverables are acceptable. These methods and deliverables can also be integrated into Microsoft Word, PowerPoint, and other design and development tools. Many of these tools are open source or widely available. Use the latest version.

1) Design:
    a. Microsoft Accessible design toolkit
    b. eBay Accessibility Annotations
    c. CVSHealth Web Accessibility Annotation Kit
    d. CVSHealth Accessibility Annotation Kit for iOS
    e. Colour Contrast Analyser
    f. WebAIM Contrast Checker
    g. Any similar design toolkit that fits well in the ADO's ways of working
2) Development:
    a. Pa11y
    b. Oobee (formerly known as Purple A11y)
    c. Microsoft Accessibility Insights
    d. ANDI (Accessible Name and Description Identifier)

    e. Axe Developer Tools
    f. WebAIM WAVE
3) Testing (at code check-in time or weekly):
    a. HHS Accessibility Conformance Checklists
    b. Microsoft Accessibility Insights manual checks
    c. Check-in script (outlined in section 1.3)
    d. Job Access With Speech (JAWS) screen reader
    e. Non-Visual Desktop Access (NVDA) screen reader
    f. Voiceover screen reader
    g. Talkback screen reader
    h. ZoomText magnification tool
    i. Magnifier magnification tool
    j. Voice access like Dragon Naturally Speaking or Windows Speech Input
    k. Any other assistive technology the team can access (switch access, etc.)

Tools may differ depending on platform (iOS, MacOS, Android, Windows). This list is not exhaustive, and other tools may be considered in the future.

2.2.1 User Stories

We recommend incorporating people with disabilities into user stories to account for accessibility throughout the software development lifecycle. This makes compliance checks easier. Here are some examples:

- As a user who is colorblind, I want status indicators with labels or icons [in addition to color] so I can understand critical information.
- As a user with a mobility disability, I want to navigate everything by keyword so I can avoid fatigue and mouse use.
- As a user who is deaf, I want training and marketing videos to include captions or transcripts so I can access all information.

More information on user stories can be found in the suggested design tool kits.

2.2.2 Types of Design Specifications

The recommended design tools support several types of design specifications.

Basic design specifications cover keyboard order and basic accessibility metadata like name, role, and value. Color contrast and use of color also need to be reviewed with basic design specifications.

Walkthrough/storyboard specifications cover keyboard interaction and changes to metadata and the user interface (UI) throughout an interaction.

2.3 Accessibility Check-in Script

2.3.1 Purpose

This script should be run each time code is checked in. This catches issues early and reduces costly accessibility fixes later.

2.3.2 Scope

Run this script on any UI surface that has code changes related to the code that will be checked in.

2.3.3 Script Steps

1. Run an automated checker (pick one of the following based on platform compatibility) and fix all obvious errors (not investigates or expanded results).
    a. Accessibility Insights Fast Pass
    b. aXe-core
    c. WebAIM WAVE
    d. Pa11y
2. Perform basic color checks
    a. Turn on high contrast and look for:
        i. Missing borders or elements
        ii. Background pictures making text hard to read
3. Inspect the accessibility tree for anomalies (name and role mismatches, extra tree items).
4. Basic keyboarding checks:
    a. You can get everywhere (using a combination of Tab, Space, Enter, Esc, and Arrow keys).
    b. You can see focus as you move around.
    c. You're never stuck (focus trapped).
    d. Circular tabbing (using Tab, go forward from the last element to get to the first).
    e. Tab and Shift + Tab cycles are reversals of each other.
5. Basic metadata checks:
    a. Turn on screen reader with default settings and swipe or down arrow through all elements. Make sure:
        i. Every element has suitable information (not repetitive, redundant, or confusing).
        ii. Information is fully captured by the screen reader.

    iii. There are no extra elements that do not contribute to functionality, or any filler spaces.
        - Example: You may hear “space, space” on these elements.
    iv. Elements are not missed or skipped.
    v. Dynamic content or things that change on the screen are announced.
        - Example: When a notification rectangle from an application appears in Windows, screen readers can read the content (when receiving a Teams chat, for example).

b. Look at the command list for anomalies (command list keyboard shortcut is screen reader and platform specific).

Appendix Glossary

- A11yDevOps: accessibility developer operations, agile software development lifecycle processes to support meeting Section 508 accessibility.
- Agile: an approach that divides work into phases, emphasizing continuous delivery and improvement.
- Clearance Officer: CMS staffer assigned to monitor accessibility in their division.