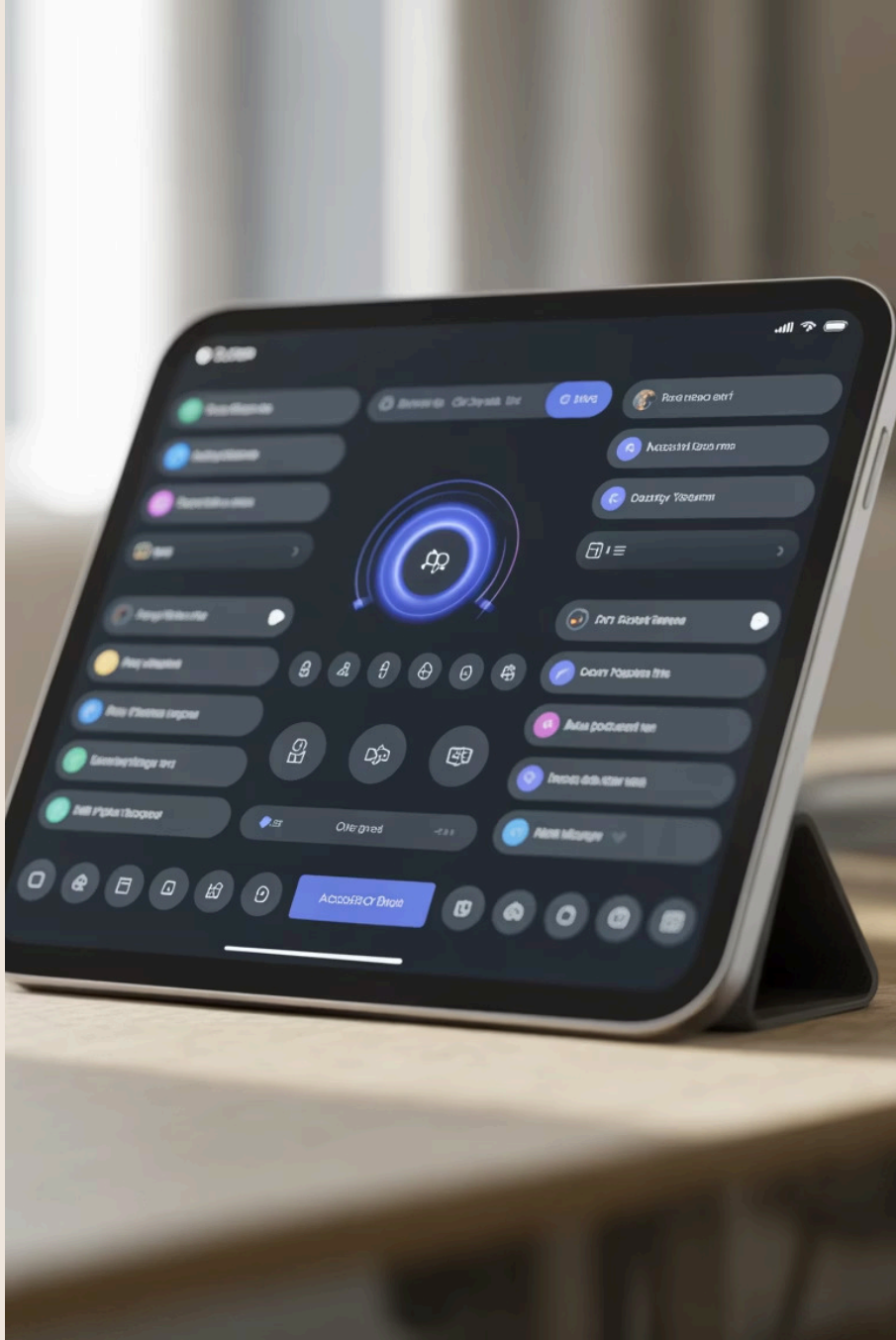
wA11y.io

wA11y Tester Toolkit

The complete field guide for accessibility quality assurance, updated for WCAG 2.2 AA. This comprehensive reference provides tools, shortcuts, defect standards, and the full compliance checklist needed for thorough testing.



Essential Validation Software

Valid testing requires specific versions and standardized settings. These tools form the foundation of professional accessibility QA work.

Screen Readers

NVDA	Windows - Standard
VoiceOver	macOS - Standard
JAWS	Windows - Enterprise

Validation Extensions

axe DevTools	Automated pass/fail checks
Accessibility Insights	Tab stops visualization
HeadingsMap	Heading hierarchy (H1-H6)

NVDA Shortcuts (Windows)

The NVDA Key is Insert or Caps Lock. Master these commands for efficient Windows-based accessibility testing.

Stop Speaking

Control

Immediately halt screen reader output

Elements List

NVDA + F7

Access Links, Headings, Landmarks

Navigation

H = Next Heading

Shift + H = Previous

B = Button | **G** = Graphic

Read All

NVDA + Down Arrow

Continuous reading from cursor

Toggle Focus Mode

NVDA + Space

Critical for form interaction testing

VoiceOver Shortcuts (macOS)

The VO Keys are Control + Option. These commands enable comprehensive macOS accessibility validation.

Stop Speaking

Control

Halt announcement immediately

The Rotor

VO + U

Navigate Headings, Links, Landmarks in organized lists

Next Item

VO + Right Arrow

Move forward through page elements sequentially

Select/Click

VO + Space

Activate focused element

Read from Top

VO + B

Start continuous reading from page beginning

The Defect Standard

Writing High-Value A11y Tickets

01

Summary

Concise, actionable description of the accessibility barrier encountered during testing

02

Severity Classification

Critical (Blocked) | **Major** (Workaround exists) | **Minor** (Cosmetic issue)

03

Environment

MUST specify exact combination: "NVDA 2024.1 + Chrome 120 on Windows 11" or "VoiceOver + Safari 17.2 on macOS Sonoma"

04

Steps to Reproduce

Numbered list with precise actions needed to replicate the defect consistently

05

Expected vs. Actual

Clear comparison showing compliant behavior versus observed failure, referencing specific WCAG criteria

Interaction & Keyboard Compliance

The "No Mouse" Test

- [ ] **Keyboard Operability (2.1.1)**

  All interactive elements are reachable and operable via Tab, Enter, Space, and arrow keys without requiring mouse input

- [ ] **No Keyboard Traps (2.1.2)**

  Focus never gets stuck in a widget or modal. Users can always navigate forward and backward freely

- [ ] **Focus Visible (2.4.7)**

  Focus indicator is always clearly visible with sufficient contrast against all backgrounds throughout navigation

- [ ] **Focus Not Obscured (2.4.11) NEW 2.2**

  Focused element is not hidden behind sticky headers, footers, or other persistent UI components

- [ ] **Target Size (2.5.8) NEW 2.2**

  All clickable areas meet minimum 24×24 CSS pixel target size for pointer activation

- [ ] **Dragging Alternatives (2.5.7) NEW 2.2**

  Sliders and drag-and-drop interfaces provide single-pointer alternatives like increment/decrement buttons

- [ ] **Consistent Help (3.2.6) NEW 2.2**

  Help/contact info needs to be available in the same location on every page

Timing & Motion Compliance

Time-based interactions and motion can create critical barriers if users cannot pause, extend, or reduce them.

- [ ] **Timing Adjustable (2.2.1)**

  Users can extend, turn off, or adjust time limits. Session timeouts provide warning and recovery options.

- [ ] **Pause, Stop, Hide (2.2.2)**

  Auto-rotating carousels, sliders, or animations can be paused. Moving or updating content does not start automatically without controls.

- [ ] **Flashing Content (2.3.1)**

  No content flashes more than three times per second. Animations do not trigger seizures or physical discomfort.

- [ ] **Motion from Interaction (2.3.3)**

  Motion triggered by scrolling or gestures can be reduced or disabled. `prefers-reduced-motion` is respected where motion is used.

Visual & Zoom Compliance

The "Display" Test

Contrast Ratios (1.4.3, 1.4.11)

Text meets 4.5:1 minimum (3:1 for large text ≥18pt or bold ≥14pt). UI components and graphical objects meet 3:1 against adjacent colors

Reflow at 400% (1.4.10)

At 1280px viewport width with 400% browser zoom, content reflows to single column without horizontal scrolling or information loss

Text Spacing (1.4.12)

Layout remains functional when users apply: 1.5× line height, 2× paragraph spacing, 0.12× letter spacing, 0.16× word spacing

Orientation (1.3.4)

Application works in both portrait and landscape orientations unless specific orientation is essential to functionality

Consistent Navigation (3.2.3)

Navigation menus, search functionality, and repeated components appear in consistent locations across all pages

Non-Text Content & Media Compliance

Non-text content and media must provide equivalent access to information for users who cannot see or hear it.

Non-Text Content (WCAG 1.1.1)

- Informative images have meaningful, descriptive alternative text
- Decorative images are hidden from assistive technology
- Icon-only buttons and controls have accessible names
- Charts, graphs, and visual data have text equivalents or summaries

Time-Based Media (WCAG 1.2.1–1.2.5)

- Video content includes accurate, synchronized captions
- Audio-only content provides transcripts
- Video with audio includes audio description or equivalent alternative
- Media controls (play, pause, volume) are keyboard accessible
- Media does not auto-play audio without user control

If information is only conveyed visually or audibly, it is not accessible.

Content & Semantics Compliance

The "Code" Test

- [ ] **Page Titles (2.4.2)**

  Every page has unique, descriptive title describing topic or purpose in `<title>` element

- [ ] **Heading Hierarchy (1.3.1)**

  Logical H1-H6 structure with no skipped levels. Single H1 per page, subsequent headings properly nested

- [ ] **Autocomplete (1.3.5)**

  Do forms support browser auto-fill (e.g., email, name)?

- [ ] **Form Labels (3.3.2)**

  All input fields have persistent, visible labels using `<label>` element or aria-label. Placeholder text alone is insufficient

- [ ] **Error Identification (3.3.1, 3.3.3)**

  Validation errors identified in text with clear descriptions and suggestions for correction. Color alone cannot convey error state

- [ ] **Status Messages (4.1.3)**

  Toast notifications and dynamic alerts announced by screen readers without moving focus using role="status" or aria-live regions

- [ ] **Redundant Entry (3.3.7) NEW 2.2**

  Can users reuse previously entered info (e.g., Shipping = Billing)?

- [ ] **Accessible Authentication (3.3.8) NEW 2.2**

  Login processes avoid cognitive tests (puzzles, character transcription) without accessible alternatives. Password paste functionality enabled

Language & Readability Compliance

Correct language identification ensures screen readers pronounce content accurately and users can understand mixed-language experiences.

Language of Page (WCAG 3.1.1)

- The default page language is defined using the `lang` attribute
- Screen readers announce content using the correct language

Language of Parts (WCAG 3.1.2)

- Language changes within content are programmatically identified
- Foreign phrases, quotes, or names use appropriate language markup
- Screen readers switch pronunciation correctly for language changes

Readability Checks (Supportive QA)

- Text avoids unnecessary jargon and abbreviations
- Abbreviations and acronyms are explained on first use
- Instructions do not rely on cultural or language-specific assumptions

If the language isn't identified correctly, assistive technology cannot interpret the content reliably.

Name, Role & Value Compliance

The "Screen Reader" Test

Custom components must expose their name, role, and state so assistive technologies can interpret and operate them correctly.

- [ ] **Name, Role, Value (WCAG 4.1.2)**
  - Interactive elements have accessible names.
  - Roles accurately reflect component behavior (e.g., button, checkbox).
  - States and values (expanded, selected, disabled) are programmatically exposed and updated.
  - Custom components announce changes without moving focus.

- [ ] **ARIA Usage Validation**
  - Native HTML elements are prioritized over ARIA.
  - ARIA is not used where native semantics suffice.
  - ARIA attributes are complete, valid, and match visible behavior.

- [ ] **Dynamic UI Verification**
  - Screen readers announce state changes in custom widgets.
  - Focus remains predictable during dynamic updates.
  - Components are fully operable via keyboard and assistive technology.

If assistive technology can’t determine what a component is or how it behaves, it isn’t accessible.

Testing Workflow Integration

Recommended Testing Sequence

1. Run automated tools (axe DevTools) to identify obvious violations
2. Perform keyboard-only navigation testing across all interactive elements
3. Validate with screen reader using realistic user scenarios and workflows
4. Test zoom, reflow, and text spacing modifications at various viewport sizes
5. Verify color contrast ratios and visual presentation across themes
6. Document findings using standardized defect template with environment details

Critical Testing Note

Automated tools detect only 30-40% of accessibility issues. Manual testing with assistive technology is essential for WCAG 2.2 AA compliance verification.

Remember: Test with real users with disabilities when possible. Their insights reveal barriers automated tools and expert testers may miss.

> This toolkit provides full tester coverage for WCAG 2.2 Level AA. Compliance is a journey, not a destination.

Accessibility testing requires continuous learning, regular updates to tools and techniques, and commitment to inclusive design principles. Each test cycle improves the digital experience for all users.

78

WCAG 2.2 AA Success Criteria

Complete standard coverage

9

New Requirements Since 2.1

Updated compliance checklist

100%

Keyboard Accessibility

Non-negotiable baseline

wA11y.io

hello@wa11y.io 🌐 wa11y.io