# Accessibility Evaluations Minimum Testing Standards

Accessibility evaluations should assess conformance with Ohio State University’s [Minimum Digital Accessibility Standards](https://go.osu.edu/mdas) (MDAS), which currently adopt the [Web Content Accessibility Guidelines 2.1](https://www.w3.org/TR/WCAG21/) at conformance level AA.

In addition, the MDAS adopt the [WAI ARIA Authoring Practices 1.1](https://www.w3.org/TR/wai-aria-practices-1.1/) to assess if custom widgets within web-based content meet expected behaviors for users of assistive technologies, and [Applying WCAG 2.0 to Non-Web Information and Communications Technologies](https://www.w3.org/TR/wcag2ict/) (WCAG2ICT) as guidance to assist in determining if other non-web-based content meets the standards.

This guidance is intended to serve as a base line of testing performed and is not intended to serve as a comprehensive guide to testing for full WCAG conformance. Vendors may supplement additional testing, at their option, to cover any testing that is not included in this base line.

## Web-based Accessibility Evaluations

This section outlines the testing that should be performed on:

* Websites
* Web-based interactive applications
* Other content presented in a web browser or browser-based view within a native application

### Test Software and Platforms

Testing should be performed using the current version of

* Windows
  + Screen reader testing performed with the current production release of NVDA and the current release of either Mozilla Firefox or Google Chrome
* The mobile responsive view, if applicable, should be tested with at least one mobile platform, e.g. the current version of Android or iOS and their respective screen readers, TalkBack or Voice Over. Testing on iOS is preferred if possible.
* Mac OSX testing is optional, but if performed should use current versions of Apple Safari and VoiceOver

### Testing Types

* Color alone should not be used to convey meaning
* Keyboard
  + Sufficient keyboard focus indication – Focus outlines should not use browser default outlines; focus should be sufficiently visible to clearly indicate keyboard focus location on a page., e.g. as a guideline, focus indicators should meet WCAG color contrast ratio requirements for text -- 3.5:1. Other acceptable techniques for sufficient indicators include sufficiently contrasting hover / focus styles that change the visual presentation or color of the focused UI element.
  + All interactive controls on a page that can be used by a pointing device (e.g. mouse) are operable via the keyboard and either
    - Conform to commonly accepted keyboard access patterns in the ARIA Authoring Practices; or
    - Sufficient instruction is provided to keyboard users when focusing the control to allow them to use the control.
* Screen Reader
  + Tables must have descriptive label, and appropriate header ID association
  + Non-text content (e.g. images) has appropriately descriptive alternative text
    - Images that are pure decoration should have empty alt (alt=””) or otherwise be hidden from screen readers
  + All interactive controls on a page that can be used by a pointing device (e.g. mouse) are operable by screen reader users, have a properly associated descriptive label, and either
    - Conform to commonly accepted keyboard interaction patterns in the ARIA Authoring Practices; or
    - Sufficient instruction is provided programmatically to screen reader users when focusing the control to allow them to use the control, and such instructions are programmatically associated with the control such that screen reader users are made aware of the instructions when focus lands on the control.
  + Semantic structures presented visually are appropriately represented programmatically
    - Lists
    - Paragraphs
    - Heading structure
      * Headings, where present, should descend sequentially appropriate to the semantics of the content
    - Landmarks and regions
      * Where multiple landmarks of a single type are present, landmarks have accessible names to disambiguate them from each other
  + The reading order of content with screen readers conveys the same meaning as the visual order.
  + Form and input fields
    - Require properly associated descriptive labels
    - Are programmatically indicated as required, when appropriate, either through the label or using an appropriate ARIA technique
    - Input validation and error messages are conveyed in an accessible way for screen reader users
* Multimedia
  + Pre-recorded audio
    - Has text transcript with speaker identification
  + Pre-recorded video
    - Has accurate, synchronized captioning with speaker identification
    - Has audio description for content presented visually that is not present in the narration
  + Live video
    - Has accurate, synchronized captioning with speaker identification
* Text and background should meet WCAG 2.0 AA color contrast ratio requirements. (Any appropriate tool using the WCAG testing formula is allowed for this testing.)
* Interfaces that modify data in storage systems, and/or make legal or financial commitments or transactions, have a method for reversing the transaction, verifying data before proceeding, or are reversible.

## Native Computer and Native Mobile Applications

For installed PC, OSX, and mobile applications, testing should generally apply [Applying WCAG 2.0 to Non-Web Information and Communications Technologies](https://www.w3.org/TR/wcag2ict/) (WCAG2ICT) to assess conformance with MDAS.

* In place of keyboard access, where a physical keyboard is not present, the platform’s screen native reader should be used to determine if UI elements are operable via assistive technology.
* Testing should be performed on the currently released latest version of the host operating system (e.g. Windows, iOS, or Android).
* In cases where the product being implemented is packaged with a pre-loaded version or custom version of a given host operating system, testing should be performed using that software stack.
* When native applications present views of web content using browser views, the web testing should be performed on the web content presented, and native application testing should be performed on non-web UI elements.

### iOS Testing

* When dynamic notifications are presented to users, notifications are presented to VoiceOver users using UIAccessibilityPostNotification.
* When Zoom is enabled and set to 2X, there is no loss of functionality or content.
  + This is intended to help meet WCAG 2.0 Success Criteria 1.4.4. Although platform level magnification software is likely considered assistive technology, and thus cannot be used to fully meet this SC, Ohio State will consider applications that do not lose content or functionality at 200% zoom to be compliant in this regard.
* Where appropriate, headings are used to denote sections of content. (Headings should visually represent hierarchy, although levels are not programmatically supported by the SDK at this time.)
* Text and background meet applicable WCAG CCR requirements based on text size.
* All informative or interactive content in the application can be focused by voiceover and is focused in an order that logically follows a reading order that preserves meaning.
* Dialogs and Menus
  + Have a properly associated descriptive label on the control that spawns the menu or dialog
  + Trap voiceover focus inside the dialog or menu
  + Are dismissible using an appropriately labelled control
  + Place focus in a logical location after user action is taken (e.g. on triggering element) when dismissed, and on top of newly loaded content when item is selected, or action is taken
* Changes in language throughout a part of an application, and the application’s base language, are programmatically determinable (or the app supports localization).
* Interactive UI elements are operable by voiceover and
  + Are focused in a logical order that matches an appropriate reading order
  + Operation or navigation of controls do not trap focus except when such behavior is expected and an accessible mechanism exists to exit the trap
  + In addition, one of the following is true regarding the interactive controls
    - A standard UIKit element is used
    - A custom control with appropriate accessibility attributes which convey name, role, and value is implemented, and instructions on using the control are programmatically available (e.g. through the use of an accessibility hint)

### Android Apps

* When dynamic notifications are presented to users, they are conveyed to TalkBack users using live regions/Toast notifications, as appropriate.
* When Magnification is enabled and set to 2X, there is no loss of functionality or content
  + This is intended to help meet WCAG 2.0 Success Criteria 1.4.4. Although platform level magnification software is likely considered assistive technology, and thus cannot be used to fully meet this SC, Ohio State will consider applications that do not lose content or functionality at 200% zoom to be compliant in this regard.
* Where appropriate, headings are used to denote sections of content, and they are used properly to denote hierarchy and relationships (Android P or later).
* Text and background meet applicable WCAG CCR requirements based on text size.
* All informative or interactive content in the application can be focused by Talkback and is focused in an order that logically follows a reading order that preserves meaning.
* Dialogs and Menus
  + Have a properly associated descriptive label on the control that spawns the menu or dialog
  + Trap TalkBack focus inside the dialog or menu
  + Are dismissible using an appropriately labelled control
  + Place focus in a logical location after user action is taken (e.g. on triggering element), when dismissed, and on top of newly loaded content where item is selected or action is taken
* Role, name, and value of interactive elements are programmatically determinable.
  + When native platform elements are used, they are provided with the requisite accessibility attributes.
  + When custom elements are implemented, must convey role/name/state/value, and manage focus, if applicable, based on action taken.