# 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/WCAG20/) at levels A and 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 the latest completed version of [Applying WCAG 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.

Note: Ohio State will notify any vendors awarded in this RFP when any changes are made to the MDAS or minimum testing requirements.

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 additional testing that is not included in this base line and should describe their testing proposal as Ohio State will consider these additional tests in scoring vendor’s responses.

## Web-based accessibility evaluations

This section outlines the testing that should be performed on:

* Websites
* Web-based interactive applications, including the portions of desktop and mobile applications that are built using web technologies
* Other content, including, but not limited to documents and videos, 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 Chrome
* The mobile responsive view, if applicable, should be tested with at least one mobile platform using the best supported browser and screen reader combination. For example, the current version of Android with Chrome and TalkBack or iOS with Safari and VoiceOver.
* macOS testing is optional and not required unless the application or website has a prominent Mac user base (20%+).

### Testing Types

1. Color alone should not be used to convey meaning
2. Keyboard
   1. 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. Other acceptable techniques for sufficient indicators include sufficiently contrasting hover / focus styles that change the visual presentation or color of the focused UI element.
   2. All interactive controls on a page that can be used by a pointing device (e.g. mouse) are operable via the keyboard and either
      1. Conform to commonly accepted keyboard access patterns in the ARIA Authoring Practices
      2. Or sufficient visual instruction is provided to keyboard users when focusing the control to allow them to use the control
   3. When keyboard shortcuts are implemented using only letter, punctuation, number, or symbol characters, either a mechanism exists to turn off the shortcut, a mechanism exists to remap the shortcut to include one or more non-printable keyboard keys (e.g., Ctrl, Alt), or the shortcut is only active when the component that uses it has focus.
   4. When keyboard focus triggers additional content to become visible and then hidden
      1. Except when the additional content does not obscure/replace other content or communicates an input error, a keyboard mechanism is available to dismiss the additional content without moving focus
      2. The additional content remains visible until focus is removed, the user dismisses it, or the information is no longer valid
3. Mouse
   1. When functionality uses multipoint or path-based gestures, there is also a way to perform the action using a single pointer unless it would fundamentally change the information or functionality of the content and the information and functionality cannot be achieved in another way that would conform
   2. For single pointer functionality, either the pointer down event is not used to execute the function, the user can abort or undo the function, the up event reverses any outcome of the preceding down event, or completing the function on the down event is essential
   3. When receiving and then removing pointer hover triggers additional content to become visible and then hidden
      1. Except when the additional content does not obscure/replace other content or communicates an input error, a keyboard mechanism is available to dismiss the additional content without removing hover
      2. The pointer can be moved over the additional content without the additional content disappearing
      3. The additional content remains visible until hover is removed, the user dismisses it, or the information is no longer valid
4. Screen reader
   1. Tables must have descriptive label, and appropriate header ID association
   2. Non-text content (e.g. images) have appropriately descriptive alternative text
      1. Images that are pure decoration should have empty alt (alt=””) or otherwise be hidden from screen readers
   3. 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
      1. Conform to commonly accepted keyboard interaction patterns in the ARIA Authoring Practices
      2. 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
   4. Semantic structures presented visually are appropriately represented programmatically
      1. Lists
      2. Paragraphs
      3. Heading structure
         1. Headings, where present, should descend sequentially appropriate to the semantics of the content
      4. Landmarks and regions
         1. Where multiple landmarks of a single type are present, landmarks have accessible names to disambiguate them from each other
   5. The reading order of content with screen readers conveys the same meaning as the visual order
   6. When a message is added to the page that provides information to the user on the success or results of an action, on the waiting state of an application, on the progress of a process, or on the existence of errors, if the message does not receive keyboard focus, it is announced to screen reader users
   7. Form and input fields
      1. Require properly associated descriptive labels
      2. Are programmatically indicated as required when appropriate either through the label or using an appropriate ARIA technique
      3. Input validation and error messages are conveyed in an accessible way for screen reader users
5. Voice Control
   1. User interface elements with labels that include text or images of text have a name that contains the text that is presented visually
6. Multimedia
   1. Pre-recorded audio
      1. Has text transcript with speaker identification
   2. Pre-recorded video
      1. Has accurate, synchronized captioning with speaker identification
      2. Has audio description for content presented visually that is not present in the narration
   3. Live video
      1. Has accurate, synchronized captioning with speaker identification
7. Text and background should meet WCAG 2.1 AA color contrast ratio requirements, any appropriate tool using the WCAG testing formula is allowed for this testing
8. Parts of graphics that are required for understanding the content as well as all non-user agent defined user interface element states should meet WCAG 2.1 AA color contrast ratio requirements, any appropriate tool using the WCAG testing formula is allowed for this testing
9. Interfaces that modify data in storage systems, make legal or financial commitments or transactions have a method for reversing the transaction, verifying data before proceeding, or are reversable.
10. Content does not restrict its view and operation to a single display orientation, such as portrait or landscape, unless it would fundamentally change the information or functionality of the content and information and functionality cannot be achieved in another way that would conform.
11. Form inputs that collect information about the user use the appropriate autocomplete attributes, see [Input Purposes for user interface components](https://www.w3.org/TR/WCAG21/#input-purposes).
12. No loss of content or functionality occurs by setting all of the following and by changing no other style property: Line height (line spacing) to at least 1.5 times the font size; Spacing following paragraphs to at least 2 times the font size; Letter spacing (tracking) to at least 0.12 times the font size; Word spacing to at least 0.16 times the font size.
13. When the screen is 320px wide and 256px tall:
    1. No content or functionality is lost, obscured, or cut off
    2. Scrolling in two dimensions is not required except for parts of the content which require two-dimensional layout for usage or meaning

## Native Computer and native mobile Applications

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

* In addition to keyboard testing, the platform’s native screen reader should be used to determine of 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), and latest version of Assistive Technologies
* 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
* All text can be increased to at least 200% using the text size settings in the operating system without loss of content or functionality
* Where appropriate, headings are used to denote sections of content and the appropriate heading level is used to reflect the heading’s visual hierarchy
* 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, on top of newly loaded content where 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)
* Content does not restrict its view and operation to a single display orientation, such as portrait or landscape, unless it would fundamentally change the information or functionality of the content and information and functionality cannot be achieved in another way that would conform.
* When functionality uses multitouch or path-based gestures, there is also a way to perform the action using a single touch unless it would fundamentally change the information or functionality of the content and the information and functionality cannot be achieved in another way that would conform
* When functionality can be operated by motion of device or the intentional gesturing by the user, except for functionality that is operated by movement through space (i.e., geolocation) or functionality that is specifically created to use device sensor date (i.e., pedometer)
  + The functionality is also operable through accessibility supported interfaces (i.e., touch, keyboard, voice)
  + The motion actuation function can be disabled to prevent accidental actuation
* 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 conveys 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
* All text can be increased to at least 200% using the text size settings in the operating system without loss of content or functionality
* 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
* Content does not restrict its view and operation to a single display orientation, such as portrait or landscape, unless it would fundamentally change the information or functionality of the content and information and functionality cannot be achieved in another way that would conform.
* When functionality uses multitouch or path-based gestures, there is also a way to perform the action using a single touch unless it would fundamentally change the information or functionality of the content and the information and functionality cannot be achieved in another way that would conform
* When functionality can be operated by motion of device or the intentional gesturing by the user, except for functionality that is operated by movement through space (i.e., geolocation) or functionality that is specifically created to use device sensor date (i.e., pedometer)
  + The functionality is also operable through accessibility supported interfaces (i.e., touch, keyboard, voice)
  + The motion actuation function can be disabled to prevent accidental actuation
* 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, on top of newly loaded content where item is selected or action is taken
* Role, name, 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, manage focus if applicable based on action taken