

# Steps to Evaluate Accessibility for Interactive GIS Applications

*These steps can be used to test and evaluate common accessibility issues for GIS web applications. This is not a complete checklist for testing ADA Title II / Section 508 compliance, refer to the [WCAG 2.1 AA Success Criteria](#) for the full list of requirements.*

## General Considerations for Evaluating Accessibility

*These points may be helpful to analyze GIS apps, but are not specific pass / fail steps.*

### Consider purpose of GIS Application

- How do users access information?
- What is the intended audience of this application?
- What questions can be asked and answered with the data?
- Do all layers or data attributes need to be included?

### Consider importance of GIS Application

- How many people use this application per month?
- Is this application the only way for users to access critical information?

### Consider structural components of GIS application

- Is there an interactive map element?
  - Most interactive maps have [inherent accessibility limitations \[1\]](#)
- Does the GIS application have separate user interface panels?
  - User interface components should meet all accessibility standards
  - User interface should provide [alternative access to map data \[2\]](#)
- Is the application full screen, scrolling page or [embedded in a web page](#)?
  - Scaling requirements may be challenging with full screen applications
  - Some layouts with Esri applications are more responsive for small screens
  - Consider the context of the final web page with embedded applications
- Was the application created with a depreciated platform?
  - Old Esri App builders will have limitations for resolving accessibility issues
  - Esri Dashboards, Experience Builder and Story Maps are modern platforms that can create flexible GIS apps with some options to improve accessibility
- Does the application include a feedback mechanism to report accessibility issues?

## Basic GIS Accessibility for Map Elements

*Test visual clarity of map elements for low vision users. Check map controls and widgets for keyboard-only navigation.*

### Map Colors: Check map for color-blind friendly palette

- ☐ Check map color palettes for compatibility with common kinds of color blindness
  - Use [ArcGIS color blindness simulator](#) or browser extension for testing
  - Red (protanopia), green (deuteranopia) and blue-yellow (tritanoptia) are the most common forms of color blindness (Henning et. al. 2017)

### Feature Styling: Analyze symbology of map features for use of color

- ☐ Confirm that map symbology doesn't rely exclusively on color to convey meaning.
  - Use shapes, line styles and fill patterns to differentiate feature symbology
- ☐ Check if map features have 3:1 color contrast ratio with surrounding elements

### Basemap Selection: Check application for high contrast basemap

- ☐ Ensure users have an option to use a high contrast basemap
- ☐ Consider making a high contrast basemap the default setting

### Map Labels: Check map labels for clarity and contrast

- ☐ Use simple, commonly used fonts for labels
  - Verdana, Arial, Calibri, Helvetica, Tahoma, Times New Roman, etc
- ☐ Use at least 12px font size, and avoid ALL CAPS for labels
- ☐ Use a [Color Contrast Checker](#) to verify labels have a 4.5:1 text color to background contrast ratio
  - Consider using a halo for map labels to ensure 4.5:1 contrast ratio

### Keyboard Navigation: Test interactive map controls for keyboard-only use

- ☐ Verify that map element has a visible focus indicator when in focus with tab key
- ☐ When map is focused, test that map navigation shortcut keys can be used
  - Can you pan with arrow keys?
  - Can you zoom with + / - keys?
- ☐ Check if all map controls and map widgets have visible focus indicator
- ☐ Check if map controls and map widgets are operable with keyboard only controls
  - *Tab, tab+shift, enter, space, esc and arrow keys may be used*

## Basic GIS Accessibility for Application User Interface

*Test if user interface controls are accessible for sighted keyboard-only users.*

### **Automated Testing:** Run accessibility browser extension to check for warnings and errors

- ☐ Use automated tools to check for basic accessibility issues with user interface
  - [Automated tools cannot detect many issues](#) for GIS Applications [3]
  - Esri user interface components should pass all automated checks
  - Currently only WAVE and Axe DevTools are on SITSD Approved software list
  - WAVE does not work well to test Esri Applications
- ☐ Run test tool on all app pages and screen states separately
- ☐ Run test tool on screens displaying error messages
- ☐ Run test tool in direct source url for embedded applications
  - Test tools will not analyze embedded content within iframes

### **Keyboard Navigation:** Test user interface elements for keyboard-only use

- ☐ Check that all interactive UI elements have a visible focus indicator with tab key
- ☐ Verify that focus indicator does not disappear when tabbing through content
- ☐ Verify that keyboard focus does not get trapped in a loop at any point
- ☐ Check that all UI controls are focusable and usable by keyboard-only controls
  - *Tab, tab+shift, enter, space, esc and arrow keys may be used*
- ☐ If full screen popup windows are used, verify that keyboard users cannot focus or interact with elements behind the popup

### **Alternative Data Access:** Verify that all important information shown by clicking on map features is also accessible for keyboard-only users

- ☐ If map elements can be clicked to show additional information, ensure that can users access this same information easily with keyboard-only controls through an alternative interface [2]

### **Links and Buttons:** Verify that links in application are unique and descriptive

- ☐ Buttons that trigger an action indicate the action they perform
- ☐ Links are unique and describe the content they point to
- ☐ Display text for links do not have raw urls or generic text such as “[click here](#)”
  - **Example:** “[click here](#) to visit the Montana Government Website” should use more descriptive link text, such as “visit the [Montana State Website](#)”

## Image Alt Text: Verify that important images have alt text

- ☐ Use testing tool or right click and inspect an image to view alt text
- ☐ Use the [Alt Text Decision Tree](#) to determine if alt text is required for an image
  - Images and graphics that convey meaning or have a function should have alt text that describes its purpose in relation to the surrounding context
  - Alt text should include any text that is inside of an image, unless that same text is redundant and present nearby in the application
  - Decorative images, or images that contain redundant information should have empty alt text ( alt="" )
- ☐ Buttons containing only icons have an accessible label that describes the function of the button
- ☐ Logos do not need to be fully described by alt text, but should have alt text for links
  - If a logo image is used as a link, alt text should describe the link
  - **Example:** If the seal for the State of Montana is used as a link to mt.gov, the alt text for the logo could be “Montana State Website”

## Headings: Check if cascading heading levels are used to outline content

- ☐ Use WAVE tool Structure Tab to analyze heading structure of application
- ☐ Heading elements are used to outline sections of user interface
- ☐ Application has unique <h1> tag at the top of each separate page
- ☐ Cascading heading structure is used throughout application:
  - <h1> application / page title
    - <h2> main content section
      - <h3> content subsection

## Sudden Changes: Check app for unexpected content changes or jarring animations

- ☐ Application does not have rapid, unexpected changes
  - Users should know what to expect from activating user interface controls
- ☐ Application does not contain content that flashes more than 3 times per second
- ☐ Application does not contain animations over 5 seconds that cannot be paused
  - Avoid very rapid and overly lengthy map panning and zooming animations

## Intermediate GIS Accessibility

*Consider users on mobile devices and users who use magnification tools. Consider accessibility of user interface for blind screen reader users.*

*These points are more difficult to resolve with existing GIS application builders, but should be checked in evaluating accessibility for GIS apps. Many of these checks can only be resolved by editing HTML or CSS code for applications.*

### **Keyboard Focus:** Check focus and reading order of UI elements for logical sequence

- ❑ Use tab key to check that UI elements are focused in a logical order for keyboard and screen reader users
- ❑ Use a screen reader such as [NVDA](#) to traverse reading order for application
  - Arrow keys and h key may be used with NVDA to step through reading order
  - Structure of HTML code should reflect tab order and reading order

### **Landmarks:** Check if landmark elements used to outline application content

- ❑ Use automated tool to analyze landmark html elements of application
- ❑ Landmark elements are used to outline sections of user interface
  - <header> <main> <footer> <region> <aside> are examples of structural, semantic elements that provide navigational landmarks for web apps

### **Text Settings:** Test application for ability for users to change text settings

- ❑ Use [accessibility bookmarklet](#) to test that users can change text settings to 1.5 line spacing, 2.0 paragraph spacing, 0.12 letter spacing and 0.16 word spacing
  - Activating the 1.4.12 Text Spacing Bookmarklet should change text spacing
  - Users will not be able to change text settings for embedded apps in an iframe

### **Skip Links:** For complex applications, verify that skip links are present to provide quick access to important sections of the user interface

- ❑ When tabbing through the top of your application, do skip links appear that move focus to important sections of the application when activated?
  - **Example:** When using tab through the heading, a “skip to data table” link may appear that moves the keyboard focus to a table element

## Screen Reader: Test User Interface with Screen Reader Software

- ☐ Use NVDA for windows, Voice Over with Mac
- ☐ Listen to how content is presented as you use tab key to navigate UI elements
- ☐ All UI form fields have labels that are announced by a screen reader
- ☐ Important images and buttons are announced by a screen reader

## Screen Reader: Check if map elements are announced to screen reader users

- ☐ Map element has an aria-label that is announced to screen reader users
  - Aria-labels for simple maps can be similar to alt text for a static map image
  - Aria-labels for complex maps could provide instructions to direct blind users to an alternative interface to access map data

## Responsive Layout: Test if app works with different screen sizes and web browsers

- ☐ Application layout shifts to preserve content as the browser window is changed
  - Ensure that content does not overlap or that important elements do not disappear when changing window sizes
- ☐ Confirm that application works with different web browsers
  - Test GIS app with Chrome, Edge, Firefox etc

## Mobile Devices: Test application on mobile devices

- ☐ Application is usable on Android and iOS mobile devices with different browsers
- ☐ Application is functional on mobile devices in portrait and landscape modes

## Screen Magnification: Test application with browser magnification

- ☐ Application scales without hidden or overlapping content up to 200% full-page magnification on a 1280px wide window
  - All text is visible at with 200% magnification
  - Application UI only scrolls in one direction with magnification
- ☐ Application scales without hidden or overlapping content up to 400% full-page magnification on a 1280px wide window
  - Application UI can scroll in one direction with magnification
  - Full-screen apps can be embedded within an iframe with a min-height property to allow for vertical scrolling with page magnification
  - 400% magnification may not be practical for GIS apps

## Advanced GIS Accessibility

*Consider how blind users may be able to directly access and understand data on an interactive map.*

*These points may not be feasible to achieve with existing GIS software, but should be considered when evaluating accessibility for GIS apps and working towards full compliance. Unless significant work has been done to implement a fully accessible solution for blind users to navigate an interactive map, it is likely that an interactive map component is completely inaccessible for blind users. [\[1\]](#)*

### **Keyboard Navigation:** Test if map features are selectable by keyboard only controls

- ☐ If map features can be highlighted and selected by a mouse, the same actions be done with keyboard-only controls [\[2\]](#)
  - Can keyboard users “click” on a map in a similar way a mouse user can?
  - Or are all clickable interactions accessible with an alternative interface?
- ☐ Map tools used to select, measure, draw and analyze data are accessible by keyboard only controls

### **Screen Reader:** Test Map Element with Screen Reader Software

- ☐ Interactive map elements are announced by screen reader software
  - Interactive map elements should have descriptive aria-label
- ☐ Check if interactive map elements have an aria-role="application" tag
  - This allows screen reader users to use map shortcut keys when the map is focused
- ☐ Navigating the interactive map is understandable with a screen reader
  - Does alt text dynamically narrate content displayed on the map?
  - Aria-live regions may be used to implement dynamic alt text
- ☐ Blind users can understand important geographic and spatial relationships presented within the interactive map
  - Can blind users engage with map data at different extents to learn about spatial relationships between points, lines and polygons?

# End Notes

## [1] Accessibility Issues for Interactive Maps

Interactive maps have inherent limitations for accessibility. The following WCAG 2.1 AA Success Criteria have been identified as difficult to achieve with existing platforms (Biggs et al., 2025).

- [WCAG SC 1.1.1](#) Maps are primarily visually oriented. Providing an equal purpose, text alternative for maps is complex, but solutions for interactive alt text exist, such as [Audiom](#).
- [WCAG SC 1.4.11](#) Non-text contrast ratios of 3:1 between all map features may be difficult to achieve for complex maps, especially with semi-transparent and blended layers.
- [WCAG SC 2.1.1](#) Keyboard Accessibility for maps requires that all functionality of the map is operable by keyboard only controls. Interactive GIS apps often do not allow keyboard-only users to ‘click’ on a map to select or highlight a feature.

## [2] Alternative Access to Map Feature Data

Ensuring alternative access to information that can be accessed by clicking on a map is important for keyboard-only and blind users (Biggs et al., 2025).

- Tables, data lists, details / feature info widgets and csv downloads can be configured to provide alternative access to GIS data
- Some components, such as lists or feature info widgets, can serve as a keyboard-only alternative method to select or highlight data on a map
- However, these alternatives may not fully satisfy legal requirements for WCAG SC 1.1.1 or SC 2.1.1.
  - For example, a csv download of GIS data does not serve the **equivalent purpose** of **clicking on an interactive map** with a mouse to open a popup displaying additional information. Manually searching through a table with keyboard-only controls is more cumbersome and less convenient than accessing the same information directly from a map.

## [3] Limitations Using Automated Accessibility Checker Tools

Automated tools such as WAVE and Axe DevTools can only analyze HTML code generated for GIS applications, and will not detect many important issues. Automated tools cannot

analyze contents of interactive maps, images, videos or iframes. Manual testing is required to test keyboard accessibility, focus order, contrast of map elements, responsive layouts and reading order for screen reader software. Accessibility Insights for Web is a helpful tool to guide through both automated and manual checks for generic web applications.

## References:

- 2025, W3C Web Accessibility Initiative, [How to Meet WCAG 2.1 Quick Reference](#)
- 2025 Biggs, Brandon et al, [Systematically Evaluating Digital Map Tools Based on the WCAG](#)
- 2024, [Accessibility Guide for Interactive Web Maps](#), Minnesota
- 2024, [Colorado GIS Mapping Accessibility Guidelines](#), Colorado
- 2017, Henning, Sabine et al, [Accessible Web Maps for Visual Impaired Users: Recommendations and Example Solutions](#)
- [Esri Accessibility Resources](#)
- [Esri Accessibility Documentation Website](#)