



Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile

W3C First Public Working Draft 26 February 2015

This version:

<http://www.w3.org/TR/2015/WD-mobile-accessibility-mapping-20150226/>

Latest published version:

<http://www.w3.org/TR/mobile-accessibility-mapping/>

Latest editor's draft:

<http://w3c.github.io/Mobile-Allv-TF-Note/>

Editors:

Kim Patch, Redstart Systems

Jeanne Spellman, W3C

Kathy Wahlbin, Interactive Accessibility

Copyright © 2015 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). W3C [liability](#), [trademark](#) and [document use](#) rules apply.

Abstract

This document, “Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile” describes how the Web Content Accessibility Guidelines (WCAG) 2.0 [WCAG20] and its principles, guidelines, and success criteria can be applied to mobile web content, mobile web apps, native apps, and hybrid apps using web components inside native apps. It provides informative guidance, but does not set requirements. It also highlights the relevance of the User Agent Accessibility Guidelines 2.0 [UAAG20] in the mobile context.

This document is intended to become a Working Group Note and is part of a series of technical and educational documents published by the [W3C Web Accessibility Initiative \(WAI\)](#).

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <http://www.w3.org/TR/>.

This document is a First Public Working Draft by the Mobile Accessibility

Task Force (Mobile A11Y TF) operating under the terms of its [Work Statement](#) under the joint coordination and review of the [Web Content Accessibility Guidelines Working Group](#) (WCAG WG) and the [User Agent Accessibility Guidelines Working Group](#) (UAWG), which is part of the [Web Accessibility Initiative](#) (WAI) of the [World Wide Web Consortium](#) (W3C). This document is intended to become a [W3C Note](#).

Feedback on this draft is essential to the success of this guidance. The Mobile Accessibility Task Force asks in particular:

1. Is this document helpful in understanding the applicability of WCAG 2.0 and UAAG 2.0 to the mobile environment?
2. Is the format of this information helpful for designers, developers and testers of content that can be viewed with mobile devices and in mobile apps? Is it useful for policymakers?
3. In Appendix A, is listing relevant existing WCAG 2.0 techniques helpful for mobile content and mobile app developers?
4. Are there additional accessibility needs in the mobile environment related to the WCAG principles that we should address?
5. Have we sufficiently explained why keyboard interface and modality independent controls are needed in the mobile environment?

To comment on this document, send email to public-mobile-a11y-tf@w3.org ([subscribe](#), [archives](#)) or [file an issue in Github](#). Comments are requested by 26 March 2015. In-progress updates to the document may be viewed in the [publicly visible editors' draft](#).

WCAG 2.0 is a stable web standard. Comments on this document will not affect WCAG 2.0 wording.

Publication as a First Public Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). The group does not expect this document to become a W3C Recommendation. W3C maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the Web Content Accessibility Guidelines Working Group and also maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the User Agent Accessibility Guidelines Working Group; those pages also include instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

This document is governed by the [1 August 2014 W3C Process Document](#).

Table of Contents

1. [Introduction](#)
 - 1.1 [WCAG 2.0 and Mobile Content/Applications](#)
 - 1.2 [Other W3C-WAI Guidelines Related to Mobile](#)

- 1.2.1 [UAAG 2.0 and Accessible Mobile Browsers](#)
 - 1.2.2 [ATAG 2.0 and Accessible Mobile Authoring Tools](#)
- 2. [Mobile accessibility considerations primarily related to Principle 1: Perceivable](#)
 - 2.1 [Small Screen Size](#)
 - 2.2 [Zoom/Magnification](#)
 - 2.3 [Contrast](#)
- 3. [Mobile accessibility considerations primarily related to Principle 2: Operable](#)
 - 3.1 [Keyboard Control for Touchscreen Devices](#)
 - 3.2 [Touch Target Size and Spacing](#)
 - 3.3 [Touchscreen Gestures](#)
 - 3.4 [Device Manipulation Gestures](#)
 - 3.5 [Placing buttons where they are easy to access](#)
- 4. [Mobile accessibility considerations related primarily to Principle 3: Understandable](#)
 - 4.1 [Changing Screen Orientation \(Portrait/Landscape\)](#)
 - 4.2 [Consistent Layout](#)
 - 4.3 [Positioning important page elements before the page scroll](#)
 - 4.4 [Grouping operable elements that perform the same action](#)
 - 4.5 [Provide clear indication that elements are actionable](#)
 - 4.6 [Provide instructions for custom touchscreen and device manipulation gestures](#)
- 5. [Mobile accessibility considerations related primarily to Principle 4: Robust](#)
 - 5.1 [Set the virtual keyboard to the type of data entry required](#)
 - 5.2 [Provide easy methods for data entry](#)
 - 5.3 [Support the characteristic properties of the platform](#)
- A. [WCAG Techniques that apply to mobile](#)
- B. [UAAG 2.0 Success Criteria that apply to mobile](#)
- C. [Acknowledgments](#)
- D. [References](#)
 - D.1 [Informative references](#)

1. Introduction

This section is non-normative.

This document provides informative guidance (but does not set requirements) with regard to interpreting and applying Web Content Accessibility Guidelines (WCAG) 2.0 [WCAG20] to web and non-web mobile content and applications.

While the World Wide Web Consortium (W3C)'s W3C Web Accessibility Initiative (WAI) is primarily concerned with web technologies, guidance for web-based technologies is also often relevant to non-web technologies. The W3C-WAI has published the Note [Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies \(WCAG2ICT\)](#) to provide authoritative guidance on how to apply WCAG to non-web technologies such as mobile native applications. The current document is a mobile-specific extension of this effort.

W3C [Mobile Web Initiative](#) Recommendations and Notes pertaining to mobile technologies also include the [Mobile Web Best Practices](#) and the [Mobile](#)

[Web Application Best Practices](#). These offer general guidance to developers on how to create content and applications that work well on mobile devices. The current document is focused on the accessibility of mobile web and applications to people with disabilities and is not intended to supplant any other W3C work.

1.1 WCAG 2.0 and Mobile Content/Applications

“Mobile” is a generic term for a broad range of wireless devices and applications that are easy to carry and use in a wide variety of settings, including outdoors. Mobile devices range from small handheld devices (e.g. feature phones, smartphones) to somewhat larger tablet devices. The term also applies to “wearables” such as “smart”-glasses, “smart”-watches and fitness bands, and is relevant to other small computing devices such as those embedded into car dashboards, airplane seatbacks, and household appliances.

While mobile is viewed by some as separate from “desktop/laptop”, and thus perhaps requiring new and different accessibility guidance, in reality there is no absolute divide between the categories. For example:

- many desktop/laptop devices now include touchscreen gesture control,
- many mobile devices can be connected to an external keyboard and mouse,
- web pages utilizing responsive design can transition into a “mobile” screen size even on a desktop/laptop, and
- mobile operating systems have been used for laptop devices.

Furthermore, the vast majority of user interface patterns from desktop/laptop systems (e.g. text, hyperlinks, tables, buttons, pop-up menus, etc.) are equally applicable to mobile. Therefore, it’s not surprising that a large number of existing WCAG 2.0 techniques can be applied to mobile content and applications (see Appendix A). Overall, WCAG 2.0 is highly relevant to both web and non-web mobile content and applications.

That said, mobile devices do present a mix of accessibility issues that are different from the typical desktop/laptop. The “Discussion of Mobile-Related Issues” section, below, explains how these issues can be addressed in the context of WCAG 2.0 as it exists or with additional best practices. All the advice in this document can be applied to mobile web sites, mobile web applications, and hybrid web-native applications. Most of the advice also applies to native applications (also known as “mobile apps”).

Note: WCAG 2.0 does not provide testable success criteria for some of the mobile-related issues. The work of the Mobile Accessibility Task Force has been to develop techniques and best practices in these areas. When the techniques or best practices don’t map to specific WCAG success criteria, they aren’t given a sufficient, advisory or failure designation. This doesn’t mean that they are optional for creating accessible web content on a mobile platform, but rather that they cannot currently be assigned a designation. The Task Force anticipates that some of these techniques will be included as sufficient or advisory in a

potential future iteration of WCAG.

The current document references existing WCAG 2.0 Techniques that apply to mobile platform (see Appendix A) and provides new best practices, which may in the future become WCAG 2.0 Techniques that directly address emerging mobile accessibility challenges such as small screens, touch and gesture interface, and changing screen orientation.

1.2 Other W3C-WAI Guidelines Related to Mobile

1.2.1 UAAG 2.0 and Accessible Mobile Browsers

The User Agent Accessibility Guidelines (UAAG) 2.0 [[UAAG2](#)] is meant for the developers of user agents (e.g. web browsers and media players), whether for desktop/laptop or mobile operating systems. A user agent that follows UAAG 2.0 will improve accessibility through its own user interface, through options it provides for rendering and interacting with content, and through its ability to communicate with other technologies, including assistive technologies.

To assist developers of mobile browsers, the [UAAG 2.0 Reference](#) support document contains numerous mobile examples. These examples are also available in a separate list of [mobile-related examples](#), maintained by the [User Agent Accessibility Guidelines Working Group \(UAWG\)](#).

1.2.2 ATAG 2.0 and Accessible Mobile Authoring Tools

The Authoring Tool Accessibility Guidelines (ATAG) 2.0 [[ATAG2](#)] provides guidelines for the developers of authoring tools, whether for desktop/laptop or mobile operating systems. An authoring tool that follows ATAG 2.0 will be both more accessible to authors with disabilities (Part A) and designed to enable, support, and promote the production of more accessible web content by all authors (Part B).

To assist developers of mobile authoring tools, the [Implementing ATAG 2.0](#) support document contains numerous mobile authoring tool examples.

Discussion of Mobile-Related Issues

2. Mobile accessibility considerations primarily related to Principle 1: Perceivable

2.1 Small Screen Size

Small screen size is one of the most common characteristics of mobile devices. While the exceptional resolution of these screens theoretically enables large amounts of information to be rendered, the small size of the screen places practical limits on how much information people can actually view at one time, especially when magnification is used by people with low vision.

Some best practices for helping users to make the most of small screens

include

- Minimizing the amount of information that is put on each page compared to desktop/laptop versions by providing a dedicated mobile version or a responsive design:
 - a dedicated mobile version contains content tailored for mobile use. For example, the content may contain fewer content modules, fewer images, or focus on important mobile usage scenarios.
 - a responsive design contains content that stays the same, but CSS stylesheets are used to render it differently depending on the viewport width. For example, on narrow screens the navigation menus may be hidden until the user taps a menu button.
- Providing a reasonable default size for content and touch controls (see "B.2 Touch Target Size and Spacing") to minimize the need to zoom in and out for users with low vision.
- Adapting the length of link text to the viewport width.
- Positioning form fields below, rather than beside, their labels (in portrait layout)

2.2 Zoom/Magnification

A variety of methods allow the user to control content size on mobile devices with small screens. At the browser level these methods are generally available to assist a wide audience of users. At the platform level these methods are available as accessibility features to serve people with visual impairments or cognitive disabilities.

The methods include the following:

- OS-level features
 - Set default text size (typically controlled from the Display Settings) Note: System text size is often not supported by mobile browsers.
 - Magnify entire screen (typically controlled from the Accessibility Settings). Note: Using this setting requires the user to pan vertically and horizontally.
 - Magnifying lens view under user's finger (typically controlled from the Accessibility Settings)
- Browser-level features
 - Set default text size of text rendered in the browser's viewport
 - Reading mode that renders main content at a user-specified text size
 - Magnify browser's viewport (typically "pinch-zoom"). Note: Using this setting requires the user to pan vertically and horizontally.
 - Note: Some browsers have features that might modify this type of magnification (e.g. re-flowing the content at the new magnification level, overriding author attempts to prevent pinch-zoom).

The WCAG 2.0 success criterion that is most related to zoom/magnification

is

- 1.4.4 Resize text (Level AA)

SC 1.4.4 requires text to be resizable without assistive technology up to 200 percent. To meet this requirement content must not prevent text magnification by the user.

The following methods might be used:

- Ensure that the browser pinch zoom is not blocked by the page's viewport meta element so that it can be used to zoom the page to 200%. Restrictive values for user-scalable and maximum-scale attributes of this meta element should be avoided. Note: Relying on full viewport zooming (e.g. not blocking the browser's pinch zoom feature) requires the user to pan horizontally as well as vertically. While this technique meets the success criteria it is less usable than supporting text resizing features that reflow content to the user's chosen viewport size. It is best practice to use techniques that support text resizing without requiring horizontal panning.
- Support for system fonts that follow platform level user preferences for text size.
- Provide on-page controls to change the text size.

Accessibility features geared toward specific populations of people with disabilities fall under the definition of assistive technology adopted by WCAG and thus cannot be relied upon to meet the success criteria. For example, a platform-level zoom feature that magnifies all platform content and has features to specifically support people with low vision is likely considered an assistive technology.

2.3 Contrast

Mobile devices are more likely than desktop/laptop devices to be used in varied environments including outdoors, where glare from the sun or other strong lighting sources is more likely. This scenario heightens the importance of use of good contrast for all users and may compound the challenges that users with low vision have accessing content with poor contrast on mobile devices.

The WCAG 2.0 success criteria related to the issue of contrast are:

- 1.4.3 Contrast (Minimum) (Level AA) which requires a contrast of at least 4.5:1 (or 3:1 for large-scale text) and
- 1.4.6 Contrast (Enhanced) (Level AAA) which requires a contrast of at least 7:1 (or 4.5:1 for large-scale text).

SC 1.4.3. allows for different contrast ratios for large text. Allowing different contrast ratios for larger text is useful because larger text with wider character strokes is easier to read at a lower contrast. This allows designers more leeway for contrast of larger text, which is helpful for content such as titles. The ratio of 18-point text or 14-point bold text described in the SC 1.4.3 was judged to be large enough to enable a lower contrast ratio for web pages displayed on a 15-inch monitor at 1024x768 resolution with a 24-inch viewing distance. Mobile device content is viewed on smaller screens and in different conditions so this allowance for lessened contrast on large text must be considered carefully for mobile apps.

For instance, the default point size for mobile platforms might be larger than the default point size used on non-mobile devices. When determining which contrast ratio to follow, developers should strive to make sure to apply the lessened contrast ratio only when text is roughly equivalent to 1.2 times bold or 1.5 times (120% bold or 150%) that of the default platform size. Note, however, that the use of text that is 1.5 times the default on mobile platforms does not imply that the text will be readable by a person with low vision. People with low vision will likely need and use additional platform level accessibility features and assistive technology such as increased text size and zoom features to access mobile content.

3. Mobile accessibility considerations primarily related to Principle 2: Operable

3.1 Keyboard Control for Touchscreen Devices

Mobile device design has evolved away from built-in physical keyboards (e.g. fixed, slide-out) towards devices that maximize touchscreen area and display an on-screen keyboard only when the user has selected a user interface control that accepts text input (e.g. a textbox).

However, keyboard accessibility remains as important as ever and most major mobile operating systems do include keyboard interfaces, allowing mobile devices to be operated by external physical keyboards (e.g. keyboards connected via Bluetooth, USB On-The-Go) or alternative on-screen keyboards (e.g. scanning on-screen keyboards).

Supporting these keyboard interfaces benefits several groups with disabilities:

- People with visual disabilities who can benefit from some characteristics of physical keyboards over touchscreen keyboards (e.g. clearly separated keys, key nibs and more predictable key layouts).
- People with dexterity or mobility disabilities, who can benefit from keyboards optimized to minimize inadvertent presses (e.g. differently shaped, spaced and guarded keys) or from specialized input methods that emulate keyboard input.
- People who can be confused by the dynamic nature of onscreen keyboards and who can benefit from the consistency of a physical

keyboard.

Several WCAG 2.0 success criteria are relevant to effective keyboard control:

- 2.1.1 Keyboard (Level A)
- 2.1.2 No Keyboard Trap (Level A)
- 2.4.3 Focus Order (Level A)
- 2.4.7 Focus Visible (Level AA)

3.2 Touch Target Size and Spacing

The high resolution of mobile devices means that many interactive elements can be shown together on a small screen. But these elements must be big enough and have enough distance from each other so that users can safely target them by touch.

Best practices for touch target size include the following:

- Ensuring that touch targets are at least 9 mm high by 9 mm wide.
- Ensuring that touch targets close to the minimum size are surrounded by a small amount of inactive space.

Note: This size is not dependent on the screen size, device or resolution. Screen magnification should not need to be used to obtain this size, because magnifying the screen often introduces the need to pan horizontally as well as vertically, which can decrease usability.

3.3 Touchscreen Gestures

Many mobile devices are designed to be primarily operated via gestures made on a touchscreen. These gestures can be simple, such as a tap with one finger, or very complex, involving multiple fingers, multiple taps and drawn shapes.

Some (but not all) mobile operating systems provide work-around features that let the user simulate complex gestures with simpler ones using an onscreen menu.

Some best practices when deciding on touchscreen gestures include the following:

- Gestures in apps should be as easy as possible to carry out. This is especially important for screen reader interaction modes that replace direct touch manipulation by a two-step process of focusing and activating elements. It is also a challenge for users with motor or dexterity impairments or people who rely on head pointers or a stylus where multi-touch gestures may be difficult or impossible to perform. Often, interface designers have different options for how to implement an action. Widgets requiring complex gestures can be difficult or impossible to use for screen reader users. Usually,

design alternatives exist to allow changes to settings via simple tap or swipe gestures.

- Activating elements via the mouseup or touchend event. Using the mouseup or touchend event to trigger actions helps prevent unintentional actions during touch and mouse interaction. Mouse users clicking on actionable elements (links, buttons, submit inputs) should have the opportunity to move the cursor outside the element to prevent the event from being triggered. This allows users to change their minds without being forced to commit to an action. In the same way, elements accessed via touch interaction should generally trigger an event (e.g. navigation, submits) only when the touchend event is fired (i.e. when all of the following are true: the user has lifted the finger off the screen, the last position of the finger is inside the actionable element, and the last position of the finger equals the position at touchstart).

Another issue with touchscreen gestures is that they might lack onscreen indicators that remind people how and when to use them. For example, a swipe in from the left side of the screen gesture to open a menu is not discoverable without an indicator or advisement of the gesture. See [Touchscreen gesture instructions](#).

3.4 Device Manipulation Gestures

In addition to touchscreen gestures, many mobile operating systems provide developers with control options that are triggered by physically manipulating the device (e.g. shaking or tilting). While device manipulation gestures can help developers create innovative user interfaces, they can also be a challenge for people who have difficulty holding or are unable to hold a mobile device.

Some (but not all) mobile operating systems provide work-around features that let the user simulate device shakes, tilts, etc. from an onscreen menu.

Therefore, even when device manipulation gestures are provided, developers should still provide touch and keyboard operable alternative control options.

- 2.1.1 Keyboard (Level A)

Another issue with control via device manipulation gestures is that they might lack onscreen indicators that remind people how and when to use them. See [Touchscreen gesture instructions](#).

3.5 Placing buttons where they are easy to access

Mobile sites and applications should position interactive elements where they can be easily reached when the device is held in different positions.

When designing mobile web content and applications many developers attempt to optimize use with one hand. This can benefit people with disabilities who may only have one hand available, however, developers should also consider that an easy-to-use button placement for some users might cause difficulties for others (e.g. left- vs. right-handed use, assumptions about thumb range of motion). Therefore, flexible use should always be the goal.

Some (but not all) mobile operating systems provide work-around features that let the user temporarily shift the display downwards or sideways to facilitate one-handed operation.

4. Mobile accessibility considerations related primarily to Principle 3: Understandable

4.1 Changing Screen Orientation (Portrait/Landscape)

Some mobile applications automatically set the screen to a particular display orientation (landscape or portrait) and expect that users will respond by rotating the mobile device to match. However, some users have their mobile devices mounted in a fixed orientation (e.g. on the arm of a power wheelchair).

Therefore, mobile application developers should try to support both orientations. If it is not possible to support both orientations, developers should ensure that it is easy for all users to change the orientation to return to a point at which their device orientation is supported.

Changes in orientation must be programmatically exposed to ensure detection by assistive technology such as screen readers. For example, if a screen reader user is unaware that the orientation has changed the user might perform incorrect navigation commands.

4.2 Consistent Layout

Components that are repeated across multiple pages should be presented in a consistent layout. In responsive web design, where components are arranged based on device size and screen orientation, web pages within a particular view (set size and orientation) should be consistent in placement of repeated components and navigational components. Consistency between the different screen sizes and screen orientations is not a requirement under WCAG 2.0.

For example:

- A Web site has a logo, a title, a search form and a navigation bar at the top of each page; these appear in the same relative order on each page where they are repeated. On one page the search form is missing but the other items are still in the same order. When the Web site is viewed on a small screen in portrait mode, the navigation bar is collapsed into a single icon but entries in the drop-down list that appears when activating the icon are still in

the same relative order.

- A Web site, when viewed on the different screen sizes and in different orientations, has some components that are hidden or appear in a different order. The components that show, however, remain consistent for any screen size and orientation.

The WCAG 2.0 success criteria that are most related to the issue of consistency are:

- 3.2.3 Consistent Navigation (Level AA)
- 3.2.4 Consistent Identification (Level AA)

4.3 Positioning important page elements before the page scroll

The small screen size on many mobile devices limits the amount of content that can be displayed without scrolling.

Positioning important page information so it is visible without requiring scrolling can assist users with low vision and users with cognitive impairments.

If a user with low vision has the screen magnified only a small portion of the page might be viewable at a given time. Placing important elements before the page scroll allows those who use screen magnifiers to locate important information without having to scroll the view to move the magnified area. Placing important elements before the page scroll also makes it possible to locate content without performing an interaction. This assists users that have cognitive impairments such as short-term memory disabilities. Placing important elements before the page scroll also helps ensure that elements are placed in a consistent location. Consistent and predictable location of elements assists people with cognitive impairments and low vision.

4.4 Grouping operable elements that perform the same action

When multiple elements perform the same action or go to the same destination (e.g. link icon with link text), these should be contained within the same actionable element. This increases the touch target size for all users and benefits people with dexterity impairments. It also reduces the number of redundant focus targets, which benefits people using screen readers and keyboard/switch control.

The WCAG 2.0 success criterion that is most related to grouping of actionable elements is:

- 2.4.4 Link Purpose (In Context) (Level A)
- 2.4.9 Link Purpose (Link Only) (Level AA)

For more information on grouping operable elements, see [H2: Combining adjacent image and text links for the same resource](#) technique.

4.5 Provide clear indication that elements are actionable

Elements that trigger changes should be sufficiently distinct to be clearly distinguishable from non-actionable elements (content, status information, etc). Providing a clear indication that elements are actionable is relevant for web and native mobile applications that have actionable elements like buttons or links, especially in interaction modes where actionable elements are commonly detected visually (touch and mouse use). Interactive elements must also be detectable by users who rely on a programmatically determined accessible name (e.g. screen reader users).

Visual users who interact with content using touch or visual cursors (e.g. mice, touchpads, joysticks) should be able to clearly distinguish actionable elements such as links or buttons. Existing interface design conventions are aimed at indicating that these visual elements are actionable. The principle of redundant coding ensures that elements are indicated as actionable by more than one distinguishing visual feature. Following these conventions benefits all users, but especially users with vision impairments.

Visual features that can set an actionable element apart include shape, color, style, positioning, text label for an action, and conventional iconography.

Examples of distinguishing features:

1. Conventional shape: Button shape (rounded corners, drop shadows), checkbox, select rectangle with arrow pointing downwards
2. Iconography: conventional visual icons (question mark, home icon, burger icon for menu, floppy disk for save, back arrow, etc)
3. Color offset: shape with different background color to distinguish the element from the page background, different text color
4. Conventional style: Underlined text for links, color for links
5. Conventional positioning: Commonly used position such as a top left position for back button (iOS), position of menu items within left-aligned lists in drop-down menus for navigation

The WCAG 2.0 success criteria do not directly address issue of clear visual indication that elements are actionable but are related to the following success criteria:

- 3.2.3 Consistent Navigation (Level AA)
- 3.2.4 Consistent Identification (Level AA)

4.6 Provide instructions for custom touchscreen and device manipulation gestures

The ability to provide control via custom touchscreen and device manipulation gestures can help developers create efficient new interfaces. However, for many people, custom gestures can be a challenge to discover, perform and remember.

Therefore, instructions (e.g. overlays, tooltips, tutorials, etc.) should be provided to explain what gestures can be used to control a given interface and whether there are alternatives. To be effective, the instructions should, themselves, be easily discoverable and accessible. The instructions should also be available anytime the user needs them, not just on first use, though on first use they may be made more apparent through highlighting or some other mechanism.

These WCAG 2.0 success criteria are relevant to providing instructions for gestures:

- 3.3.2 Labels or Instructions (Level A)
- 3.3.5 Help (Level AAA)

5. Mobile accessibility considerations related primarily to Principle 4: Robust

5.1 Set the virtual keyboard to the type of data entry required

On some mobile devices, the standard keyboard can be customized in the device settings and additional custom keyboards can be installed. Some mobile devices also provide different virtual keyboards depending on the type of data entry. This can be set by the user or can be set to a specific keyboard. For example, using the different HTML5 form field controls (see [Method Editor API](#)) on a website will show different keyboards automatically when users are entering in information into that field. Setting the type of keyboard helps prevent errors and ensures formats are correct but can be confusing for people who are using a screen reader when there are subtle changes in the keyboard.

5.2 Provide easy methods for data entry

Users can enter information on mobile devices in multiple ways such as on-screen keyboard, Bluetooth keyboard, touch, and speech. Text entry can be time-consuming and difficult in certain circumstances. Reduce the amount of text entry needed by providing select menus, radio buttons, check boxes or by automatically entering known information (e.g. date, time, location).

5.3 Support the characteristic properties of the platform

Mobile devices provide many features to help users with disabilities interact with content. These include platform characteristics such as zoom, larger fonts, and captions. The features and functions available differ depending on the device and operating system version. For example, most platforms have the ability to set large fonts, but not all applications honor it for all text. Also, some applications might increase font size but not wrap text, causing horizontal scrolling.

A. WCAG Techniques that apply to mobile

[WCAG 2.0 Techniques that Apply to Mobile](#)

B. UAAG 2.0 Success Criteria that apply to mobile

[UAAG Mobile Accessibility Examples](#)

C. Acknowledgments

Participants in the Mobile Accessibility Task Force:

Kathleen Anderson
Jonathan Avila
Tom Babinszki
Matthew Brough
Michael Cooper (WCAG WG Staff Contact)
Gavin Evans
Detlev Fischer
Alistair Garrison
Marc Johlic
David MacDonald
Kim Patch (TF Facilitator – UAAG WG)
Jan Richards
Mike Shebanek
Brent Shiver
Alan Smith
Jeanne Spellman (UAAG WG Staff Contact)
Henny Swan
Peter Thiessen

Kathleen Wahlbin (TF Facilitator – WCAG WG)

Chairs of the WCAG WG and the UAAG WG:

Jim Allan (UAAG WG), Texas School for the Blind and Visually Impaired
Andrew Kirkpatrick (WCAG WG), Adobe Systems
Joshue O Connor (WCAG WG), National Council for the Blind of Ireland (NCBI)

This publication has been funded in part with Federal funds from the U.S. Department of Education, National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR) under contract number ED-0SE-10-C-0067. The content of this publication does not necessarily reflect the views or policies of the U.S. Department of Education, nor does mention of trade names, commercial products, or organizations imply

endorsement by the U.S. Government.

D. References

D.1 Informative references

[UAAG20]

James Allan; Kelly Ford; Kimberly Patch; Jeanne F Spellman. [User Agent Accessibility Guidelines \(UAAG\) 2.0](#). 25 September 2014. W3C Working Draft. URL: <http://www.w3.org/TR/UAAG20/>

[WCAG20]

Ben Caldwell; Michael Cooper; Loretta Guarino Reid; Gregg Vanderheiden et al. [Web Content Accessibility Guidelines \(WCAG\) 2.0](#). 11 December 2008. W3C Recommendation. URL: <http://www.w3.org/TR/WCAG20/>